



**UNIVERSIDAD NACIONAL DE INGENIERIA
RECINTO UNIVERISTARIO 'SIMÓN BOLÍVAR'
FACULTAD DE ELECTROTENENCIA Y
COMPUTACION**

TRABAJO MONOGRÁFICO

**“Sistema Administrativo de Seguro
Para la Corredurías”**

**PARA OPTAR EL TÍTULO DE
*INGENIERO EN COMPUTACIÓN***

ELABORADO POR:

***Br. José Francisco Fonseca Hernández
Br. Christian del Rosario Rocha García***

TUTOR

Ing. Gabriel Rafael Lacayo Saballos

MANAGUA, NICARAGUA

Julio, 2017

DEDICATORIA A:

Dios, nuestro tierno padre celestial, por darme sabiduría, su amor, bendición y porque gracias a él hemos logrado la culminación de este trabajo y cada una de la metas en nuestras vidas.

Sara Hernández, Sebastián Fonseca, Ruth García y Ernesto Rocha, nuestros padres, por ser las personas más maravillosas y esforzados de este mundo, porque por sus luchas e incondicional amor hemos logrado ser las personas que ahora somos.

Nuestros hijos (Jeshua, Montserrat y Franshesco) pilares de nuestras vidas que nos impulsan día a día para sigan adelante.

J.Fonseca, C.Rocha

RESUMEN

El presente trabajo monográfico plantea el desarrollo de un sistema de Información de escritorio para el control de ventas de pólizas, Control de Siniestros, Pagos de Comisiones, pagos de otros ingresos en las diferentes corredurías, utilizando un lenguaje de programación de bajo costo y un gestor de Base de Datos.

Se emplea Visual .NET como lenguaje de programación, SQL Server 2012 como gestor de Base de Datos y se propone publicar en un servidor Cliente-Servidor sobre plataforma Windows Server, ya que SQL server y Visual .NET corren en Sistemas Operativos Microsoft.

SASC¹ está conformado por un conjunto de Pantallas dinámicas, a través de las cuales se permite el ingreso, modificación, actualización y consulta de la Información en la base de datos del sistema.

Este trabajo monográfico fue desarrollado según las necesidades planteadas por el usuario y siguiendo las técnicas y metodologías propuestas por la ingeniería de software.

La metodología adoptada se inspiró en los métodos orientados a objetos y aprovechando las ventajas de UML como notación y lenguaje de modelado, aunque el proceso implementado fue definido por los autores. En este proceso se realizó análisis (identifica el “que” es lo que el software resolverá), diseño (se concentra en “como” deberá trabajar la aplicación) y la implementación o construcción del sistema.²

¹ Sistema Administrativo de Seguros para las Corredurías.

INDICE

1. Introducción.....	1
2. Objetivos.....	3
2.1 Objetivo General.....	3
2.2 Objetivos Específicos	3
3. Justificación.....	4
4. Marco teórico	6
4.1 Arquitecturas Cliente-Servidor.....	6
4.1.1 Arquitecturas.....	6
4.1.2 Servidores.	7
4.1.3 Servidores de Archivos.	7
4.1.4 Servidores de Base de Datos.	7
4.1.5 Servidores de Transacciones.....	8
4.1.6 Servidores Groupware.	8
4.1.7 Servidores de Aplicaciones de Objetos.	8
4.2 Servidores de Aplicaciones WEB.....	9
4.3 Modelo en dos capas (Two-Tier Model).	9
4.1.8 4.3.1 Ventajas Modelo En Dos Capas.....	10
4.1.9 4.3.2 Desventajas Modelo En Dos Capas.	10
4.4 Requerimientos de Hardware y Software	11
5. Herramientas de desarrollo y lenguajes de modelados	12
5.1 Visual .NET	12
5.1.1 5.1.1 Características.....	12
5.1.2 5.1.2 Entorno de Desarrollo.....	14
5.1.3 5.1.3 Ventajas	17
5.1.4 5.1.4 Desventajas.....	18
5.2 Base de Datos SQL Server 2012 R2.....	18
5.1.5 Características	19
5.1.6 Desventajas.....	20
5.3 UML.....	20
5.4 Metodología de Desarrollo	21
5.1.7 5.4.1 Los ciclos de desarrollo del marco metodológico Blue Watch	24
5.4.2 Características del marco metodológico Blue Watch.....	25

5.4.3 Estructura y componentes del marco metodológico Blue Watch.	26
6. Análisis Del Sistema	28
6.1.1 Técnicas e instrumentación de recolección de datos.	28
6.1.2 Reconocimiento de Requerimientos	32
7. ANALISIS DE LA INFORMACIÓN	34
7.1 Estudio Técnico Tecnológico	34
7.2 Etapas Y Tiempo Del Desarrollo De Software	36
7.3 Estudio Administrativo Y Organizativo	39
7.4 Estudio De Costos	44
8. Diseño Del Sistema	46
8.1 Especificación de Requerimientos	46
8.2 Diagrama de Actividades.....	52
8.3 Desarrollo de Diagrama Entidad – Relación.	55
9. Diseño de los Formularios del Sistema.	58
10. Pruebas del Sistema.....	62
11. Conclusión.....	70
12. Recomendaciones	71
13. Bibliografía.....	72
14. Anexos	74
15. Formato de encuesta	75
16. Ejemplos de pólizas.	78

Índice de Tablas.

Tabla 1: Requerimientos Funcionales.....	33
Tabla 2: Requerimientos No Funcionales.....	33
Tabla 3: Etapas y Tiempo de Desarrollo del Software.....	38
Tabla 4: Estructura Administrativa y Organizativa.....	42
Tabla 5: Mano de Obra.....	44
Tabla 6: Materia Prima.....	45
Tabla 7: Ingresar Cliente.....	48
Tabla 8: Modificar Cliente.....	49
Tabla 9: Eliminar Cliente.....	50
Tabla 10: Entidades Potenciales.....	55
Tabla 11: Pruebas Funcionales.....	63
Tabla 12: Pruebas de Comportamiento.....	65
Tabla 13: Pruebas de Aceptación.....	67

Índice de ilustraciones.

Ilustración 1: Esquema de Arquitectura Cliente Servidor clásica.....	10
Ilustración 2: Caso de Uso (Sistema de Registro y Control).....	43
Ilustración 3: Caso de Uso General (Procesos a Realizar).....	47
Ilustración 4: Caso de Uso (Manejo de Clientes).....	48
Ilustración 5: Ingreso de Cliente.....	52
Ilustración 6: Modificar Cliente.....	53
Ilustración 7: Eliminar de Cliente.....	54
Ilustración 8: Diagrama Inicial con Todas las Entidades Resultantes de los Requerimientos.....	56
Ilustración 9: Diagrama Entidad – Relación (Sistema SASC ³).....	57
Ilustración 10: Ingreso de Clave de Acceso (Sistema SASC).....	58
Ilustración 11: Ingreso de Tasa de Cambio del día (Sistema SASC).....	59
Ilustración 12: Submenú Transacciones.....	59

³ Sistema Administrativo de Seguros para las Corredurías.

Ilustración 13: Pantalla de Ingreso de Póliza Automotor.....	60
Ilustración 14: Pantalla de Ingreso de Póliza Responsabilidad Civil de Licencia de Conducir.....	60
Ilustración 15: Pantalla de Ingreso de Abonos a Pólizas.....	61

1. INTRODUCCIÓN

El mercado asegurador se encuentra actualmente muy fragmentado en distintas compañías que apuestan fuertemente por la utilización de tecnología que permita el manejo y control de la información de forma más eficiente, haciendo uso de los sistemas de información.

Las empresas actuales dependen cada vez más de las tecnologías de la información para alcanzar sus objetivos corporativos, lo cual es un recurso imprescindible para cualquier empresa que quiera ofrecer a sus clientes una mejor calidad, ventaja, eficacia o valor agregado de cada una de las decisiones de la organización.

El uso de un sistema informático posibilita la gestión de toda la base de datos de los usuarios que permita obtener el máximo rendimiento e información, además los clientes son cada vez más exigentes y el mercado se encuentra en un proceso de saturación, donde el cliente dispone de más información sobre los distintos productos de otras empresas de seguro, por lo que hemos creado un sistema que nos permita gestionar de manera estratégica y dinámica nuestra relación con ellos.

El presente trabajo monográfico está enfocado en el desarrollo de un Sistema Administrativo de Seguro para las Corredurías, denominado SASC por sus siglas en español.

El desarrollo de este sistema es altamente seguro y privado, utilizando altos estándares de seguridad y encriptación. El usuario administrador del sistema

podrá definir permisos a nivel de usuario, permitiendo indicar si un usuario puede ver y modificar datos, o sólo ver datos, o no tener ningún tipo de acceso a algunos de los formularios e informes del sistema. Para esto existe un catálogo de usuarios con clave o contraseña, de manera que se puede identificar qué usuario agrego un registro específico en las ventanas a las que tenga acceso completo.

Este software será de mucha utilidad para las corredurías, permitiendo una mejor atención a los usuarios, así como un mejor cumplimiento de los compromisos de informar a la SIBOIF y sus clientes, permitiendo disponer de información en tiempo real.

2. OBJETIVOS

2.1 Objetivo General

- Implementar un sistema automatizado de información que permita agilizar y controlar las transacciones de los procesos en las áreas técnicas y administrativas de la correduría de seguros “La Nacional S, A” durante el ciclo de vida de la suscripción de una póliza y la atención de sus reclamos, permitiendo así una mejor manejo de la Información, Consulta y Estadísticas que faciliten la toma de decisiones.

2.2 Objetivos Específicos

- Levantar los requerimientos relacionados con las operaciones que se necesitan automatizar.
- Analizar los requerimientos levantados de forma cuantitativa y cualitativa, teniendo como resultado un documento de análisis el cual debe ser presentado a las áreas correspondientes de la correduría para su aprobación.
- Diseñar un prototipo del sistema a partir del documento de análisis aprobado por las áreas correspondientes de la correduría, utilizando el programa UML como herramienta de modelado.
- Desarrollar un sistema automatizado de información teniendo en cuenta el diseño del prototipo, como una solución adecuada que permita el registro, almacenamiento y actualización de la información tanto de los clientes, pólizas y reclamos y que muestre la información registrada y procesada a través de consultas y reportes.
- Implementar el sistema automatizado de información por etapas acordadas entre los usuarios de la correduría y los desarrolladores del sistema.

3. JUSTIFICACIÓN

En los últimos años el sector de seguros en Nicaragua ha experimentado sólido crecimiento. Pero detrás de ese dinamismo hay más de 400 subagentes que mueven más del 50 por ciento del negocio (Valenzuela, 2013). Dichas empresas carecen de un sistema de información que permita registrar esta gran cantidad de información, generando retraso en la atención a los clientes, en la entrega de informes a la Superintendencia de Bancos, al igual que desorden en el manejo de la información.

El objetivo principal de realizar este sistema consiste en que se podrá agilizar las operaciones que realizan las corredurías, que se ve limitado cuando tiene que recopilar toda la información posible de sus clientes. La forma de trabajar del propietario queda obsoleta ya que no existe una base de datos unificada y el empleado no puede obtener toda la información del cliente en el momento deseado.

La principal tarea es poder tener toda la información unificada para que de esta manera, utilizarla de una manera ágil y eficiente, minimizando el tiempo invertido por el empleado y suministrando una información veraz y oportuna. Es ahí donde el uso de un sistema informatizado nos tiene que posibilitar gestionar toda la base de datos de nuestro cliente. Para llevar a cabo dicha tarea, se han analizado un conjunto de lenguajes informáticos, con la finalidad de poder escoger el más idóneo para las distintas tareas que se plantean, teniendo en cuenta la creación de un conjunto de módulos que se encargan de procesar los datos de los clientes, productos, compañías, pólizas y recibos.

Esta flexibilidad de operación del sistema aportara grandes beneficios a las corredurías, fortalecerá el logro de sus objetivos y mejorará la satisfacción de los clientes y usuarios.

El paso de la elaboración de informes en hojas de cálculos de Microsoft Excel a Informes generados automáticamente por el sistema asegura la credibilidad y veracidad de los datos presentado a la superintendencia de Bancos y otra instituciones financieras, como llevar más controlado los pagos de siniestros y comisiones de los vendedores.

Todo esto contribuirá a:

- Mejora del servicio al cliente.
- Mejoras en la toma de decisiones y en la planificación.
- Mejoras en la productividad.
- Construcción de innovaciones de negocio.

4. MARCO TEÓRICO

4.1 Arquitecturas Cliente-Servidor

Cliente-Servidor.

Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores".

Un proceso envía un mensaje que representa una petición. El mensaje se entrega a un receptor, que procesa la petición y envía un mensaje como respuesta. En secuencia, la réplica puede disparar posteriores peticiones, que llevan a nuevas respuestas, y así, sucesivamente.

La arquitectura cliente-servidor sustituye a la arquitectura monolítica en la que no hay distribución, tanto a nivel físico como a nivel lógico. (Molina, 2013)

4.1.1 Arquitecturas.

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente dentro de la organización.

Debemos señalar que para seleccionar el modelo de una arquitectura, hay que partir del contexto tecnológico y organizativo del momento y, que la

arquitectura Cliente/Servidor requiere una determinada especialización de cada uno de los diferentes componentes que la integran. (Ary, 2003)

4.1.2 Servidores.

Es cualquier recurso de cómputo dedicado a responder a los requerimientos del cliente. Los servidores pueden estar conectados a los clientes a través de redes LANS o WANS, para proveer de múltiples servicios a los clientes y ciudadanos tales como impresión, acceso a bases de datos, fax, procesamiento de imágenes, etc. (Ary, 2003)

4.1.3 Servidores de Archivos.

- Solicitud de registros de archivo
- Servicio de información muy primitivo
- Muchos mensajes para obtener información
- Útiles para compartir archivos a lo largo de red.
- Son responsables para crear repositorios compartidos (documentos, imágenes, planos de ingeniería y otros objetos de datos grandes de la red). (Viera, 2014)

4.1.4 Servidores de Base de Datos.

- Cliente pasa como mensajes solicitudes escritas en SQL (lenguaje de consulta estructurado).
- Servidor emplea su propio poder de procesamiento para encontrar los datos pedidos, en vez de entregar toda la información.
- Proveen información para la toma de decisiones a través de reportes y consultas.

- Tienen una función vital en el almacenamiento masivo de información.
(Viera, 2014)

4.1.5 Servidores de Transacciones.

- Cliente llama por medio de un motor de base de datos de SQL a procedimientos. remotos que residen en el servidor
- Procedimientos remotos (almacenados): conjunto de instrucciones SQL.
- Intercambio de la red consiste en un solo mensaje de solicitud una respuesta.
- Instrucciones SQL tienen éxito o fracasan como unidad
- Transacciones: Son instrucciones SQL agrupadas. (Viera, 2014)

4.1.6 Servidores Groupware.

- Se encarga de la administración de información como texto, imágenes, correo electrónico, tablero y boletines electrónicos. (Viera, 2014)

4.1.7 Servidores de Aplicaciones de Objetos.

- Está escrito como un conjunto de objetos de comunicación.
- Los objetos clientes se comunican con objetos del servidor mediante un intermediario de solicitud de objetos (ORB object request broker).
- El cliente invoca un método sobre un objeto remoto del servidor, el ORB localiza una instancia de esa clase de objeto en el servidor, llama al método invocado y entrega el resultado al objeto al cliente.
- Los servidores de objetos deben dar soporte para que exista concurrencia (solicitudes simultáneas) y comunicación.
- Un ejemplo de Servidor de Aplicaciones de Objetos son los servidores de aplicaciones JAVA (CORBA).

- Microsoft tiene su propio ORB llamado modelo de objetos de componentes distribuidos (DCOM, distributed component object model).

Un modelo de componentes es el ActiveX. (Viera, 2014)

4.2 Servidores de Aplicaciones WEB.

- Los servidores WEB entregan documentos (Páginas WEB) a los clientes cuando estos los piden.
- Clientes y servidores se ponen en contacto a través de un protocolo denominado HTTP (protocolo de transferencia de).
- HTTP: Protocolo que define un conjunto simple de comandos.
- Servidores Web de Objetos: Un comienzo en servidores de WEB de objetos son los Applets de JAVA, son muy parecidos a los servidores de aplicaciones de objetos. (Viera, 2014)

4.3 Modelo en dos capas (Two-Tier Model).

En una arquitectura cliente/servidor clásica tenemos dos "capas" (two-tier):

- Una donde está el cliente que implementa la interface.
- Otra donde se encuentra el gestor de bases de datos que trata las peticiones recibidas desde el cliente. (Ary, 2003)

La lógica de la aplicación se encuentra por tanto repartida entre el cliente y servidor (Ilustración 1).

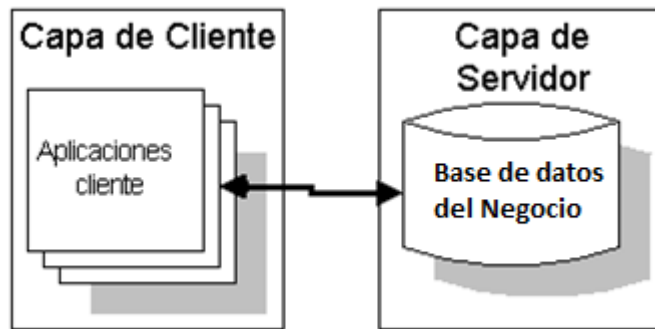


Ilustración 1: Esquema de Arquitectura Cliente Servidor clásica.

4.1.8 4.3.1 Ventajas Modelo En Dos Capas.

- El desarrollo de aplicaciones en un ambiente de dos capas es mucho más rápido que en ambientes anteriores, pero no es necesariamente más rápido que con el nuevo ambiente de tres capas.
- Las herramientas para el desarrollo con dos capas son robustas y evaluadas.
- Las técnicas de prototipo se emplean fácilmente.
- Las soluciones de dos capas trabajan bien en ambientes no dinámicos estables, pero no se ejecutan bien en organizaciones rápidamente cambiantes. (Lopez Carrasco, 2011)

4.1.9 4.3.2 Desventajas Modelo En Dos Capas.

- Los ambientes de dos capas requieren control excesivo de las versiones y demandan esfuerzo de distribución de la aplicación cuando se les hacen cambios. Esto se debe al hecho de que la mayoría de la aplicación lógica existe en la estación de trabajo del cliente.
- La seguridad del sistema en un diseño de dos capas es compleja y a menudo requiere administración de las bases de datos; esto es debido al

número de dispositivos con acceso directo al ambiente de esas bases de datos.

- Las herramientas del cliente y de la base de datos, utilizadas en diseños de dos capas, constantemente están cambiando. La dependencia a largo plazo de cualquier herramienta, puede complicar el escalamiento futuro o las implementaciones.
- El número usuarios máximo es de 100. Más allá de este número de usuarios se excede la capacidad de procesamiento. (Lopez Carrasco, 2011)

4.4 Requerimientos de Hardware y Software

Para este tipo de arquitectura, se requiere una máquina para la aplicación y una máquina para el servidor de base de datos, o se puede ocupar solo una máquina para la aplicación y para la base de datos.

También necesitamos un manejador de base de datos para poder editarla en caso de fallos de la aplicación, de igual manera necesitamos la aplicación con la que vamos a trabajar. (Lopez Carrasco, 2011).

5. HERRAMIENTAS DE DESARROLLO Y LENGUAJES DE MODELADOS

5.1 Visual .NET

Se trata de una plataforma para el desarrollo de software que fue lanzada por Microsoft con la finalidad de fusionar su amplio catálogo de productos, que va desde sus múltiples sistemas operativos hasta herramientas de desarrollo. Desde un punto de vista tecnológico, lo que se pretendía con la creación de .NET era poder desarrollar aplicaciones y sistemas que fueran independientes de la arquitectura física y del sistema operativo sobre el que se ejecutaran.

Desde el punto de vista comercial, .NET podría interpretarse como la alternativa por parte de Microsoft en el sector de los desarrollos web para competir con la plataforma Java de Oracle Corporation, así como con los diversos framework basados en PHP.

Al tratarse de una plataforma de propósito general, se pueden realizar tanto desarrollos web, como programas de escritorio o aplicaciones para dispositivos móviles. Además favorece el desarrollo en multiplataforma, como por ejemplo el que una misma aplicación pueda correr indistintamente en los diferentes sistemas operativos móviles como IOS, Android o Windows Phone garantizando la comunicación entre los diferentes dispositivos. (Provencio, 2003)

5.1.1 5.1.1 Características.

La plataforma Microsoft .NET está completamente basada en el paradigma de Orientación a Objetos (para más información acerca de este tema puede consultar el material de estudio de la Estrella 0 del programa)

.NET es multi-lenguaje: esto quiere decir que para poder codificar aplicaciones sobre esta plataforma no necesitamos aprender un único lenguaje específico de programación de alto nivel, sino que se puede elegir de una amplia lista de opciones. Veremos este tema con mayor detalle más adelante en la presentación.

.NET es una plataforma que permite el desarrollo de aplicaciones empresariales de misión crítica, entendiéndose por esto que permite la creación y ejecución de aplicaciones de porte corporativo que sean críticas para la operación de tipos variados de organizaciones. Si bien también es muy atrayente para desarrolladores no profesionales, estudiantes y entusiastas, su verdadero poder radica en su capacidad para soportar las aplicaciones más grandes y complejas. (jjparra, 2008)

.NET fue diseñado de manera tal de poder proveer un único modelo de programación, uniforme y consistente, para todo tipo de aplicaciones (ya sean de formularios Windows, de consola, aplicaciones Web, aplicaciones móviles, etc.) y para cualquier dispositivo de hardware (PC's, Pocket PC's, Teléfonos Celulares Inteligentes, también llamados "SmartPhones", Tablet PC's, etc.). Esto representa un gran cambio con respecto a las plataformas anteriores a .NET, las cuales tenían modelos de programación, bibliotecas, lenguajes y herramientas distintas según el tipo de aplicación y el dispositivo de hardware.

Uno de los objetivos de diseño de .NET fue que tenga la posibilidad de interactuar e integrarse fácilmente con aplicaciones desarrolladas en plataformas anteriores, particularmente en COM, ya que aún hoy existen una gran cantidad de aplicaciones desarrolladas sobre esa base.

.NET no sólo se integra fácilmente con aplicaciones desarrolladas en otras plataformas Microsoft, sino también con aquellas desarrolladas en otras plataformas de software, sistemas operativos o lenguajes de programación. Para esto hace un uso extensivo de numerosos estándares globales que son de uso extensivo en la industria, y acerca de los cuales iremos aprendiendo a lo largo del curso. Algunos ejemplos de estos estándares son XML, HTTP, SOAP, WSDL y UDDI (jjparra, 2008)

5.1.2 5.1.2 Entorno de Desarrollo.

La gama de productos de Visual Studio comparte un **único entorno de desarrollo integrado (IDE)** que se compone de varios elementos: la barra de menús, la barra de herramientas Estándar, varias ventanas de herramientas que se acoplan u ocultan automáticamente a la izquierda, en la parte inferior y a la derecha, así como en el espacio del editor. Las ventanas de herramientas, menús y barras de herramientas disponibles dependen del tipo de proyecto o archivo en el que esté trabajando.

Sistema de proyectos. Las soluciones y los proyectos contienen elementos en forma de referencias, conexiones de datos, carpetas y archivos necesarios para crear la aplicación. Un contenedor de tipo solución puede contener varios proyectos y un contenedor de tipo proyecto normalmente contiene varios elementos. Para obtener más información, vea Soluciones como contenedores y Proyectos como contenedores.

El *Solution Explorer* muestra soluciones, sus proyectos y los elementos incluidos en dichos proyectos. En el Explorador de soluciones, puede abrir

archivos para editar, agregar nuevos archivos a un proyecto y ver las propiedades de las soluciones, proyectos y elementos.

El editor y los diseñadores que utilice dependerán del tipo de archivo o documento que esté creando. El Editor de texto es el procesador de textos básico del IDE, mientras que el Editor de código es el editor de código fuente básico.

Otros editores, como el Editor CSS, el Diseñador HTML y el Diseñador de páginas Web, comparten muchas de las características del Editor de código, junto con mejoras específicas en el tipo de código o de marcado admitido.

Los editores y diseñadores normalmente tienen dos vistas: una vista de diseño gráfica y la vista de código subyacente o vista de código fuente. La vista de diseño le permite especificar la ubicación de los controles y otros elementos en la interfaz de usuario o la página web. Puede arrastrar controles desde el cuadro de herramientas y colocarlos en la superficie de diseño.

Visual Studio.NET proporciona un sólido conjunto de **herramientas de compilación y depuración**. Con las configuraciones de compilación puede seleccionar los componentes que se van a generar, excluir los que no se van a generar y determinar cómo se van a generar los proyectos seleccionados y en qué plataforma. Puede tener configuraciones de compilación para soluciones y para proyectos. Para obtener más información, vea *Cómo: Crear y editar configuraciones y Versiones de generación durante el desarrollo de una aplicación*.

Cuando genera, está comenzando el proceso de depuración. La compilación de la aplicación le ayuda a detectar errores de compilación. Estos errores pueden deberse a una sintaxis incorrecta, a palabras clave mal escritas o a divergencias entre los tipos. La Resultados (Ventana) muestra estos tipos de errores.

Herramientas de Implementación. Visual Studio.NET proporciona dos estrategias de implementación diferentes: *ClickOnce* y *Windows Installer*. Con la implementación de *ClickOnce*, publica la aplicación en una ubicación centralizada y el usuario instala o ejecuta la aplicación desde esa ubicación. La implementación de *Windows Installer* permite empaquetar la aplicación en un archivo setup.exe y distribuir ese archivo entre los usuarios; ellos ejecutan el archivo setup.exe para instalar la aplicación. Para ver una comparación detallada, consulte Elegir una estrategia de implementación.

ClickOnce permite implementar rápidamente aplicaciones mediante el Asistente para publicación.

Documentación de productos. Para obtener acceso a la Ayuda, puede presionar F1 en el IDE o hacer clic en Documentación de Visual Studio en el menú Ayuda. La documentación de la Ayuda se muestra en el explorador web. Puede utilizar la Ayuda instalada localmente o MSDN en pantalla y otros de recursos en pantalla para obtener ayuda. (Díaz, 2015)

5.1.3 5.1.3 Ventajas

- **Código administrado:** El CLR realiza un control automático del código para que este sea seguro, es decir, controla los recursos del sistema para que la aplicación se ejecute correctamente.
- **Interoperabilidad multilenguaje:** El código puede ser escrito en cualquier lenguaje compatible con .Net ya que siempre se compila en código intermedio (MSIL).
- **Compilación just-in-time:** El compilador JIT incluido en el Framework compila el código intermedio (MSIL) generando el código máquina propio de la plataforma. Se aumenta así el rendimiento de la aplicación al ser específico para cada plataforma.
- **Garbage collector:** El CLR proporciona un sistema automático de administración de memoria denominado recolector de basura (garbage collector). El CLR detecta cuándo el programa deja de utilizar la memoria y la libera automáticamente. De esta forma el programador no tiene por qué liberar la memoria de forma explícita aunque también sea posible hacerlo manualmente (mediante el método `Dispose` liberamos el objeto para que el recolector de basura lo elimine de memoria).
- **Seguridad de acceso al código:** Se puede especificar que una pieza de código tenga permisos de lectura de archivos pero no de escritura. Es posible aplicar distintos niveles de seguridad al código, de forma que se puede ejecutar código procedente del Web sin tener que preocuparse si esto va a estropear el sistema.

➤ **Despliegue:** Por medio de los ensamblados resulta mucho más fácil el desarrollo de aplicaciones distribuidas y el mantenimiento de las mismas. El Framework realiza esta tarea de forma automática mejorando el rendimiento y asegurando el funcionamiento correcto de todas las aplicaciones (Provencio, 2003)

5.1.4 5.1.4 Desventajas

- Consumo elevado de recursos de la maquina al utilizar la administración del código CLI:
- La ejecución del código en otros lenguajes obligan a que estos tengan el protocolo CLS (Common Language Specification)
- La ejecución de las aplicaciones desarrolladas en .net por medio de una máquina virtual. net, requieren un mayor gasto de recursos en comparación a una aplicación que se ejecute de una manera nativa
- El manejo de la memoria a través del recolector de basura consume gran cantidad de recursos y además le resta control al usuario sobre los datos de la aplicación. (Provencio, 2003)

5.2 Base de Datos SQL Server 2012 R2.

Es un sistema de manejo de bases de datos del modelo relacional, desarrollado por la empresa Microsoft. El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de management studio) es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

Los competidores principales de SQL Server, en el mercado de las bases de datos relacionales, son productos como Oracle, MariaDB, MySQL, PostgreSQL etc.; SQL Server solo está disponible para sistemas operativos Windows de Microsoft.

Puede ser con ilustración do para utilizar varias instancias en el mismo servidor físico, la primera instalación lleva generalmente el nombre del servidor, y las siguientes - nombres específicos (con un guion invertido entre el nombre del servidor y el nombre de la instalación). (Aranda, 2016)

5.1.5 Características

- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.
- Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.
- Es común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa la base de datos (Microsoft SQL

Server), con el entorno de desarrollo (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows.

- En el manejo de SQL mediante líneas de comando se utiliza el SQLCMD, osql, o PowerShell.
- Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos.
- El tipo NUMERIC fue mejorado para ser usado como identificador de columna a partir de la versión 2008 R2. (Aranda, 2016)

5.1.6 Desventajas

En versiones de 32 bits, SQL Server usa *Address Windowing Extension (AWE)* para hacer el direccionamiento por encima de 4 GB. Esto le impide usar la administración dinámica de memoria, y sólo le permite (Aranda, 2016) alojar un máximo de 64 GB de memoria compartida. Esta limitación es exclusiva de sistemas operativos 32 bits; en sistemas operativos 64 bits, la memoria máxima que se puede direccionar en Edición Estándar es 64 Gb y en Edición Enterprise 4Tb. (Aranda, 2016)

5.3 UML

UML es una técnica para la especificación sistemas en todas sus fases. Nació en 1994 cubriendo los aspectos principales de todos los métodos de diseño antecesores y, precisamente, los padres de UML son Grady Booch, autor del método Booch; James Rumbaugh, autor del método OMT e Ivar Jacobson, autor

de los métodos OOSE y Objectory. La versión 1.0 de UML fue liberada en Enero de 1997 y ha sido utilizado con éxito en sistemas construidos para toda clase de industrias alrededor del mundo: hospitales, bancos, comunicaciones, aeronáutica, finanzas, etc

Los principales beneficios de UML son:

- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos. (Aydoo, 2015)

5.4 Metodología de Desarrollo

En esta investigación la metodología a desarrollar es la Blue WATCH que es un marco de trabajo metodológico para el desarrollo de aplicaciones empresariales de mediana complejidad y/o tamaño. Algunas típicas aplicaciones, en las que este marco metodológico se puede aplicar son las siguientes: comercio electrónico, gobierno electrónico, sistemas de información web, portales corporativos, aplicaciones para dispositivos móviles y aplicaciones colaborativas.

Un marco de trabajo metodológico para el desarrollo de software es un conjunto de modelos que describen los productos, procesos y actores que deben participar en un proyecto de desarrollo de software. Un marco metodológico se diferencia de un método de desarrollo de software en que el primero sirve de guía o patrón para la elaboración del segundo. Una premisa importante usada para diseñar Blue WATCH como marco metodológico es la conclusión dada por Boehm y Turner (2003) que establece que “es mejor construir un método gradualmente, que adaptarlo a partir de uno muy elaborado”. Blue WATCH puede ser usado como un patrón al cual, las empresas que deseen utilizarlo, le van agregando elementos metodológicos, o lo van modificando progresivamente, para crear un método propio adaptado a las características particulares de la empresa, de sus proyectos y de sus grupos de trabajo.

Blue WATCH está compuesto por tres modelos. El primero de ellos, denominado Modelo de Productos, describe el conjunto de productos intermedios y finales que se deben elaborar durante el desarrollo de cualquier aplicación empresarial de pequeña o mediana complejidad. El segundo modelo, llamado Modelo de Procesos, describe las actividades técnicas, de gestión y de soporte que se requieren ejecutar para elaborar una aplicación. Finalmente, el Modelo de Actores identifica los roles requeridos para desarrollar una aplicación y recomienda una estructura organizacional para el grupo de trabajo. Los tres modelos son patrones metodológicos adaptables y extensibles. El Modelo de Procesos es representado a través de una cadena de valor. Los procesos ubicados en la parte superior de la cadena de valor son los procesos técnicos que Blue WATCH recomienda seguir

para desarrollar una aplicación empresarial. En la parte inferior de la cadena, se ubican los procesos de gestión del proyecto y los procesos de apoyo; ambos se llevan a cabo a lo largo de la ejecución de todos los procesos técnicos. Blue WATCH está basado en el enfoque de Desarrollo de Software Guiado por Modelos – MDA (Mellor et al, 2003); enfoque mediante el cual, una aplicación se desarrolla a través de la transformación de modelos elaborados usando el lenguaje UML. Dicha transformación se apoya en herramientas CASE que facilitan la elaboración de los modelos y su conversión automática. Los modelos producidos, siguiendo el marco metodológico Blue WATCH, son transformados a lo largo de una serie de procesos técnicos del desarrollo, que van desde el Modelado del Negocio hasta las Pruebas de la Aplicación, pasando por los procesos de Desarrollo de Requisitos, Diseño Arquitectónico, Diseño Detallado y Programación & Integración.

➤ **Procesos para el desarrollo del software según marco metodológico Blue Watch**

- Modelado del Negocio.
- Desarrollo de Requisitos.
- Diseño Arquitectónico.
- Desarrollo de Versiones.
- Gestión del Proyecto.
- Gestión de Requisitos.
- Verificación & Validación (V&V).

- Gestión de la Configuración del Software (GCS).
- Gestión de Riesgos.
- Aseguramiento de la Calidad del Software (ACS).

5.1.7 5.4.1 Los ciclos de desarrollo del marco metodológico Blue Watch

El Modelo de Procesos del Blue WATCH identifica y ordena los procesos técnicos, de gestión y de soporte que se requieren para desarrollar una aplicación empresarial. Este modelo le indica, al grupo de desarrollo, cómo deben llevarse a cabo estos procesos. Este modelo es iterativo, versionado e incremental. Es iterativo porque sus procesos se ejecutan cíclicamente, es decir, se repiten un número determinado de veces hasta alcanzar el resultado deseado. Es versionado porque la aplicación se realiza, gradualmente, en varias versiones, con un ciclo para cada versión. Es incremental porque los programas, de cada versión, se desarrollan en incrementos que se van entregando al cliente gradualmente.

El modelo de procesos del Blue WATCH está inspirado en la metáfora del reloj de pulsera, que consta de un motor que hace avanzar las horas, una vez completado un ciclo de minutos; y hace avanzar los minutos, una vez completado un ciclo de segundos. Cada aplicación se desarrolla mediante un conjunto de ciclos de procesos. El ciclo de una aplicación contiene uno o más ciclos de versiones y cada uno de estos últimos contienen, a su vez, varios ciclos de iteraciones.

Cada ciclo del incremento produce una pieza de código ejecutable denominada incremento. Esta pieza de código se produce en un tiempo muy corto (1- 3

semanas). Cada incremento implementa un subconjunto de funciones de la aplicación, las cuales, generalmente, se representan mediante casos de uso en la notación UML. Un incremento produce resultados concretos para el usuario, que aunque son parciales, le permiten al usuario validar partes de la aplicación. Cada ciclo de versión produce una versión completa y operativa de la aplicación. Una versión es un producto acabado que implementa un conjunto completo de funciones y puede ser utilizada inmediatamente, por sus usuarios, una vez que ella ha sido instalada en su plataforma de operación. Tal como se plantea en la figura anterior, una aplicación es un producto de software que evoluciona, a lo largo del tiempo, a través de una serie de versiones; cada una de las cuales es producida por un ciclo de versión.

5.4.2 Características del marco metodológico Blue Watch

- Es completo: Cubre el ciclo de desarrollo de la aplicación e integra procesos técnicos del desarrollo de software con los de soportes y gestión del proyecto.
- Es extensible y adaptable: Sus componentes se pueden extender para incorporar otros procesos, productos, prácticas o actores propios de la cultura de desarrollo de una empresa o de las características particulares de un proyecto.
- Es reutilizable: Una vez extendido o adaptado, el marco de trabajo puede reutilizarse como un método estándar para el desarrollo de todos los proyectos de software de una empresa.

- Produce un conjunto mínimo de documentos: El método produce un número reducido de documentos que no afectan el proceso de mantenimiento de la aplicación.
- Es balanceado: Proporciona balance entre agilidad y disciplina. Combina prácticas ágiles con procesos y prácticas disciplinadas, algunas de las establecidas en el modelo CMMI ⁴y los cuerpos de conocimiento SWEBOK ⁵y PMBOK⁶.

5.4.3 Estructura y componentes del marco metodológico Blue Watch.

Blue WATCH está compuesto por tres modelos fundamentales:

- El modelo de productos: Describe los productos intermedios y finales que el marco metodológico propone elaborar durante el desarrollo de una aplicación empresarial.
- El modelo de procesos: Describe los procesos técnicos, gerenciales y de soporte que el grupo de trabajo debe seguir durante la ejecución de un proyecto de desarrollo.
- El modelo de actores: Identifica los roles necesarios para elaborar una aplicación empresarial y sus respectivas responsabilidades dentro del proceso de desarrollo de la aplicación. Este modelo propone, también, una

⁴ Desarrollo de sistemas dinámicos (en inglés Dynamic Systems Development Method o DSDM).

⁵ Software Engineering Body of Knowledge

⁶ *Project Management Body of Knowledge*

estructura que sirve para organizar al grupo de trabajo que desarrollará la aplicación.

Cada modelo es un patrón metodológico extensible y adaptable. Las empresas o grupo de desarrollo, que estén interesadas en reutilizar el marco metodológico, deben adaptar cada modelo a sus necesidades y requisitos particulares.

(Humboldt, 2015)

6. ANÁLISIS DEL SISTEMA

El uso de un sistema informatizado tiene que posibilitar gestionar toda la base de datos de nuestro usuario para poderle sacar el máximo rendimiento, además los clientes son cada vez más exigentes y el mercado se encuentra en un proceso de saturación, donde el cliente dispone de más información sobre los distintos productos. Los usuarios se ven limitados, cuando tienen que obtener toda la información posible de sus clientes. La forma de trabajar de la correduría queda obsoleta ya que no existe una base de datos unificada y el usuario no puede obtener todos los datos del cliente en el momento deseado esto genera la necesidad de orientarse hacia el cliente para poder establecer una relación y poder construir estrategias a nivel de marketing, almacenaje de datos, tratamiento analítico e informes.

6.1.1 Técnicas e instrumentación de recolección de datos.

La recolección de datos es una etapa de acción, observación y medición, en la cual es obtenida la información para poder efectuar el análisis respectivo conducente a la población del problema planteado, o a determinar si la solución tentativamente propuesta es respaldada por los hechos. El proceso de recolección de datos consiste en seleccionar un instrumento de medición, aplicarlo y luego preparar las mediciones obtenidas. (Busot, 1991)

En este estudio, el proceso de recolección de datos se realizó a través de técnicas de observación mediante encuesta. La observación por encuesta consiste en obtener datos de interés para la investigación por medio del interrogatorio a los sujetos del universo en estudio. (Cohen, 2000)

El instrumento básico de la investigación mediante encuesta es el cuestionario, el cual, consiste en un conjunto de preguntas respecto a una o más variables, según la investigación, las cuales deben ser respondidas por los elementos de la población o muestra o la muestra de estudio.

➤ **El Cuestionario** este procedimiento requiere que los sujetos de la encuesta puedan ser reunidos en un mismo lugar y en una misma hora. La aplicación de un cuestionario de este tipo puede llevarse a cabo de dos maneras: haciendo una explicación previa a la tarea de responder por parte de los encuestados o bien entregando directamente los cuestionarios para que sean respondidos por cada uno de los miembros del grupo encuestado. (Cohen, 2000)

En el presente trabajo monográfico, se utilizó la modalidad del cuestionario simple. El cuestionario simple diseñado para este estudio fue elaborado de acuerdo con los objetivos de la investigación, la variable, las dimensiones e indicadores de las mismas. En él se utilizaron preguntas cerradas, las cuales contienen categorías o alternativas de respuestas preestablecidas, en ellas se presentan las posibles respuestas y los sujetos deben responder de acuerdo con las mismas. Agregan los autores que este tipo de preguntas pueden incluir diversas modalidades, así como las alternativas múltiples y tipo escala.

La estructura del cuestionario es la siguiente, un cuestionario simple dirigido al Personal de la correeduría, conformado por quince (8) ítems, con preguntas de tipo escala, con la opción de respuesta, casi siempre, algunas veces, casi nunca y nunca.

También para este estudio se utilizó otro tipo de recolección de datos como las entrevistas con este sistema se captan tanto las opiniones como los criterios personales del encuestado y mediante ellos se profundiza en los juicios emitidos para hacer las interpretaciones convenientes:

➤ **Las Entrevistas** que consisten en una conversación entre dos personas por lo menos, en la cual uno es el entrevistador y otro es el entrevistado. Estas personas dialogan para el arreglo de un problema teniendo un propósito profesional.

Conforme al propósito profesional con que se utiliza, puede cumplir con algunas de estas funciones:

- Obtener información de individuos o grupos.
- Facilitar información.

Las entrevistas pueden ser estructuradas o formal; no estructurada o informal. (Tamayo, 2002)

En presente trabajo monográfico las entrevistas se realizaron en la correduría de seguros “**La Nacional S, A**”, con el personal encargado de obtener las estadísticas anuales y los encargados de suministrar toda la información del proceso de registro y control de clientes y pólizas. En la Fase de Análisis generalmente se aplican a Gerentes o empleados que son los que serán afectados al aplicar el sistema propuesto.

Otro tipo de instrumentos para la investigación, se trata de la captación de datos como producto de un seguimiento sistemático del hecho en estudio dentro

de su medio para conocer su conducta y comportamiento, así como las características particulares.

➤ **La Observación** es un procedimiento de recopilación de datos e Información que consiste en utilizar los sentidos para observar hechos y realidades presentes y a la gente en el contexto real en donde desarrolla normalmente sus actividades. (Tamayo, 2002)

En este estudio se observaron los diferentes procesos manuales de registros y control. En la Fase de Análisis, es el que más tiempo requiere porque necesita estar presente en el lugar donde se desarrollan y hacer la inspección a fin de contemplar todos los aspectos inherentes a su comportamiento y características de ese campo. Al observar las operaciones le proporciona al analista hechos que no podría obtener por medio de otras técnicas de recolección de datos.

Mediante los procedimientos de observación y revisión de los registros se logró detectar cada uno de los procesos que intervienen en la recopilación de la información de los datos personales de clientes, agentes y pólizas la correduría, observando que al llevarlas de esta forma lo que genera es grandes desórdenes administrativo no satisfaciendo las necesidades y requerimientos de los usuarios, debido al gran flujo de clientes supervisado por esta organización.

Ahora bien, por medio de otros instrumentos de recolección de datos como: la entrevista y los cuestionarios a los actores del proceso, se logró obtener una

evaluación sobre las actividades a realizarse, y se determinó que esta tarea cuenta con los siguientes eventos:

De Registro: aquí se llena el formato con todos los datos personales del cliente.

De Control: Se controlan todas las situaciones especiales en las que se encuentra el cliente, las ventas, reclamos y control de pagos realizados.

De Solicitud: Cuando el cliente requiera saber el estado en que se encuentran sus pólizas.

Y de Reportes: Maneja varios tipos de reportes tales como, cuantos y cuales son la cantidad de pólizas vendidas, reclamos ingresados, cantidad de clientes que se encuentra en una condición dada.

6.1.2 Reconocimiento de Requerimientos

Luego de un análisis previo del dominio de la aplicación, es importante determinar las funciones que desea el usuario, para luego definirlos como requerimientos. Mediante una encuesta aplicada a la población y muestra, se logró analizar las solicitudes y necesidades, del objeto de estudio, a las cuales se pretende responder. Por ello, se desarrolló una serie de peticiones para ser manejadas por el sistema propuesto, éstas se clasificaron en relación al flujo de información manejado por un sistema de información.

Tabla 1: Requerimientos Funcionales

Número	Requerimiento	Descripción
RF1	Almacene Información	El sistema permitirá guardar nuevos clientes, pólizas y pagos.
RF2	Permita dar de Baja Información	El sistema deberá permitir anular clientes, pólizas, pagos y usuarios.
RF3	Permita Actualizar Información	El sistema deberá permitir la modificación de clientes y pólizas.
RF4	Genere Reportes	Con la generación de reportes se podrá obtener la información de cantidad de pólizas vendidas, ingreso total obtenido por día, mes y año.

Tabla 2: Requerimientos No Funcionales

Número	Requerimiento	Descripción
RNF1	Usabilidad	Debe ser fácil de usar, con ayudas e interfaces intuitivas.
RNF2	Seguridad	El ingreso al sistema estará restringido bajo contraseñas cifradas y usuarios definidos.
RNF3	Portabilidad	El sistema debe brindar comodidad al usuario y a otras áreas.
RNF4	Multiplataforma	El sistema deberá de funcionar en distintos tipos de sistemas operativos y plataformas de hardware.
RNF5	Desempeño	El sistema no presentara problemas para su manejo e implementación.

7. ANALISIS DE LA INFORMACIÓN

7.1 Estudio Técnico Tecnológico

Durante la implementación del sistema se han realizado pruebas unitarias y funcionales para cada una de las funciones verificando el correcto funcionamiento del sistema y comprobando que cumple la funcionalidad establecida en los requisitos.

Finalizada la implementación se han realizado pruebas para comprobar el funcionamiento global de las aplicaciones utilizando diferentes datos de prueba. Se ha verificado y validado la funcionalidad global y se ha comprobado la correcta accesibilidad e iniciación de los diferentes módulos que conforman el sistema SASC.

A medida que las aplicaciones han sido probadas y evaluadas a nivel local se ha escalado la solución a diferentes entornos de pruebas hasta su implantación final en el servidor de producción. Los diferentes entornos utilizados para la realización de las pruebas tienen características hardware y software similares al entorno final, siendo el entorno de preproducción el más parecido al entorno de producción.

Desarrollo → QA o Certificación → Producción

Una vez realizada la implantación del sistema en el entorno de producción se ha efectuado la supervisión de la correcta accesibilidad, iniciación e inicialización del sistema.

Características del hardware requerido:**Pc Programador:**

- 1 procesadores intel Core 2 Duo de 2 Ghz
- 1 discos duros 160 GB Sata
- Memoria 1 GB
- 1 tarjetas de Red

Servidor:

- x3850
- Procesador: 1 X Intel® Xeon® DualCore 7110N 2.50GHz Memoria Caché: 4MB L3
- Memoria RAM: 2GB Disco Duro: 2 T
- Unidad óptica: DVD ROM
- Interfaz de red: Dual Gigabit Ethernet Fuente de poder: 2 x Power Supplies
- Arquitectura: Rack
- Garantía: 3 años
- Entrega 30 días
- Tarjeta Dialogic con capacidad para 240 puestos.

7.2 Etapas Y Tiempo Del Desarrollo De Software

El cuadro que se observa a continuación permite analizar las etapas de desarrollo junto con sus responsables y tiempo invertido en cada una de ellas.

Etapa	Responsables	Tiempo
Expresión de necesidades Esta etapa tiene como objetivo la consecución de un primer documento en que queden reflejados los requerimientos y funcionalidad que ofrecerá al usuario del sistema a desarrollar (qué, y no cómo, se va a desarrollar).	Un Ingeniero de Documentación	10 días
Especificaciones Por medio de esta etapa se obtendrá un nuevo documento que definirá con más precisión el sistema requerido por el cliente, serán necesarias sucesivas versiones del documento en que irán quedando reflejada la evolución de las necesidades del cliente	Un Ingeniero de Documentación	5 días
Análisis Planteamiento de estructura, relaciones, evolución en el tiempo, detalle de funcionalidades que van a dar una descripción clara de qué sistema vamos a construir, qué funcionalidades va a aportar y qué comportamiento va a tener.	Un Ingeniero de Documentación Dos Ingenieros de desarrollo y Soporte	30 Días

<p>Diseño</p> <p>Tras la etapa anterior ya se tiene claro que debe hacer el sistema, ahora tenemos que determinar cómo va a hacerlo, aquí se definirán en detalle entidades y relaciones de las bases de datos, se pasará de casos de uso esenciales a su definición como casos expandidos reales, se seleccionará el lenguaje más adecuado, sistema gestor de base de datos, librerías, configuración de hardware y redes.</p>	<p>Dos Ingeniero de desarrollo y Soporte</p>	<p>40 días</p>
<p>Implementación</p> <p>Llegado este punto se empieza a codificar algoritmos y estructuras de datos, definidos en las etapas anteriores, en el correspondiente lenguaje de programación y/o para un determinado sistema gestor de</p>	<p>Dos Ingeniero de desarrollo y Soporte</p>	<p>15 Días</p>
<p>Pruebas y Ajustes</p> <p>El objetivo de estas pruebas es garantizar que el sistema ha sido desarrollado correctamente, sin errores de diseño y/o programación. Es conveniente que sean planteadas al menos tanto en el ámbito de cada módulo, como de integración del</p>	<p>Un Auxiliar de pruebas y capacitación</p> <p>Dos Ingeniero de desarrollo y Soporte</p>	<p>10 Días</p>
<p>Validación</p> <p>Esta etapa tiene como objetivo la verificación de que el sistema desarrollado cumple con los requisitos expresados inicialmente por el cliente y que han dado lugar al presente proyecto</p>	<p>Un Ingeniero de Documentación</p> <p>Un Auxiliar de pruebas y capacitación</p>	<p>10 Días</p>

Mantenimiento y evolución Pequeñas operaciones tanto de corrección como de mejora de la aplicación, así como otras de mayor importancia, fruto de la propia evolución	Un Ingeniero de desarrollo y Soporte	
---	--------------------------------------	--

Tabla 3: Etapas y Tiempo de Desarrollo del Software

7.3 Estudio Administrativo Y Organizativo

Para el proyecto de desarrollar un sistema administrativo de seguros a la medida se requiere las siguientes características en el personal y en la organización del mismo:

Cargo	Perfil	Roles
Director proyecto	Un Ingeniero de sistemas líder experto en manejo de recursos tecnológicos y humanos, capacidad de resolución de problemas. Con conocimiento	Gestor de proyectos: planea, gestiona y asigna recursos, forma prioridades, coordina interacciones con clientes y usuarios y mantiene centrado al equipo de proyecto Revisor de gestión: evalúa la planificación del proyecto y los productos de trabajo de valoración del proyecto en puntos de revisión importantes del ciclo vital del proyecto.

<p>Ingeniero de Documentación</p>	<p>Un Ingeniero de sistemas con experiencia en levantamiento de información y conocimientos sobre seguros.</p>	<p>Analista de sistemas: investiga los requisitos del sistema, coordina la requisición esquematizando la funcionalidad del sistema y delimitándolo.</p> <p>Especificador de requisitos: especifica y mantiene los requisitos del sistema desarrollado y administra las solicitudes del cliente.</p> <p>Gestor de control de cambios: define y supervisa el proceso de control de cambios.</p> <p>Soporte al proceso de desarrollo: soporta el desarrollo de software y se encarga de producir materiales adicionales que necesita el producto final.</p>
-----------------------------------	--	--

<p>Ingeniero de Desarrollo y Soporte</p>	<p>Dos Ingenieros de Sistemas con experiencia en desarrollo de software sobre .Net y motor de bases de datos SQL server.</p>	<p>Arquitecto de software: incluye la promoción y la creación de soporte para las decisiones técnicas claves que restringen el diseño global y la Implementación para proyecto.</p> <p>Diseñador: dirige el diseño de una parte del sistema, dentro de las restricciones de los requisitos, arquitectura y proceso de desarrollo para el proyecto.</p> <p>Diseñador de base de datos: dirige el diseño de la estructura de almacenamiento de datos persistentes que se utilizaran en el sistema.</p> <p>Diseñador de interfaz de usuario: coordina el diseño de la interfaz de usuario. Esto incluye recopilar los requisitos de utilización y los diseños de interfaz de usuario candidata a la creación de prototipos para cumplir estos requisitos.</p> <p>Implementador: desarrolla los componentes de software y efectúa las pruebas de desarrollado para la integración en subsistemas más grandes</p> <p>Integrador dirige la planificación y la ejecución de la integración del elemento de implementaron para producir compilaciones</p> <p>Administrador del sistema: mantiene la infraestructura de desarrollo de hardware y software (instalación, configuración y copias de seguridad)</p>
--	--	--

<p>Auxiliar de pruebas y capacitación</p>	<p>Un estudiante de ingeniería de sistemas mínimo quinto semestre.</p>	<p>Gestor de pruebas: Dirige el esfuerzo de prueba global. Incluye el apoyo de calidad, la planificación, gestión de recursos y la resolución de cuestiones que impiden el esfuerzo de prueba</p> <p>Analista de pruebas: identifica y define las pruebas necesarias, supervisa el proceso de pruebas, Evalúa los resultados y la calidad global.</p> <p>Diseñador de pruebas: dirige la definición del enfoque de pruebas y garantiza la Implementación satisfactoria</p> <p>Capacitador: se encarga de transmitir el conocimiento específico sobre el software desarrollado a los diferentes actores que intercalan con la herramienta.</p>
---	--	--

Tabla 4: Estructura Administrativa y Organizativa.

Organigrama



Estructura organizativa (Ilustración 2)

Conclusión:

Se requieren para el desarrollo del sistema administrativo de seguros los siguientes recursos administrativos:

- El director del proyecto que corresponde al director del área de desarrollo de la empresa, el tiempo invertido de este recurso en el proyecto es intermitente.
- Un ingeniero de documentación de tiempo completo.
- Dos ingenieros de desarrollo de tiempo completo.
- Un auxiliar de pruebas y capacitación de tiempo completo.

Es factible contar con los recursos que se requieren para el desarrollo de la herramienta debido a que el recurso humano requerido se encuentra en este momento dentro de la compañía trabajando en otros proyectos y es posible tomarlos para el proyecto de desarrollo del sistema administrativo de seguros en el momento que se requiera.

7.4 Estudio De Costos

Partiendo de la premisa que el proyecto se va desarrollar dentro de una empresa ya consolidada en un departamento de desarrollo de software que se encuentra en actividad desde hace 4 años los costos generados adicionales por este proyecto son los siguientes:

Mano De Obra Costos Directos					
Cargo	Tiempo	Salario Mensual	Otros / mensual		Total / 6 meses
Programador 1	6 meses	\$500.000	Ir Servicio Profesional 10%	\$-50	\$2,700
Programador 2	6 meses	\$500.000	Ir Servicio Profesional 10%	\$-50	\$2,700
Ingeniero de desarrollo y soporte (Trabajador de la Empresa)	6 meses	\$ 0.000 (La empresa asigno el apoyo de este recurso)	Alimentación y Viáticos	\$60	\$ 360.00
Usuarios Lideres que manejan los procesos de la empresa	6 meses	\$ 0.000 (La empresa asigno el apoyo de este recurso)	Alimentación y Viáticos	\$ 60	\$ 360.00
Auxiliar de Pruebas	6 meses	\$ 0.000 (La empresa asigno el apoyo de este recurso)	Alimentación y Viáticos	\$ 60	\$ 360.00
Costo Total Inversión en Recurso					\$6480.00

Tabla 5: Mano de Obra.

Materia prima Costos directos				
Materia Prima	Cantidad	Valor	Otros	Valor Total
Pc's ambiente de desarrollo	2	\$824.50	Iva: \$123.67	\$1896.35
Servidor de Producción.	1	\$3,475.63	Iva: \$521.34	\$ 3996.97
Lic. SQL Server (Para un servidor)	1	\$ 850		\$ 850
Total Costos Materia Prima				\$ 6,743.32

Tabla 6: Materia Prima.

Conclusión:

Dentro de los costos incurridos en el desarrollo de sistema administrativo de seguros se encontró que los costos directos de Mano de obra se calculan por un valor de \$6480.00, estos costos corresponden únicamente a los 6 meses que dura el desarrollo del proyecto. Los costos directos de materia prima requeridos son \$ 6,743.32 dentro de los cuales se puede analizar que el mayor costo de materia prima corresponde al servidor de comunicación con la que debe interactuar el software para lograr su funcionalidad. El proyecto tiene un total de costos directos de \$13223.32

8. DISEÑO DEL SISTEMA

8.1 Especificación de Requerimientos

Luego de haber especificar los requerimientos y elaborado el prototipo usuario/sistema, se especificaron detalladamente y de manera formal o técnica los requerimientos de los usuarios del sistema, usando gráficos y modelos de casos de uso, en función con los resultados obtenidos, teniendo en cuenta las fallas y necesidades de los procesos.

Por tal motivo, las correderías se han visto en la necesidad de ir a implantar un sistema para el proceso de registro y control, donde el usuario pueda interactuar directamente con el sistema a través de la computadora, permitiendo que el registro del cliente y póliza este organizada y disponible para cuando se necesite cualquier consulta.

En el modelo funcional se identifican los actores con los diferentes procesos, quienes tienen como objetivo satisfacer de manera eficaz sus propios deseos al ser considerados los usuarios del sistema, en donde el actor principal es guiado por un patrón. El siguiente gráfico explica de forma general el Caso de Uso para ese proceso (Ver Ilustración 2).

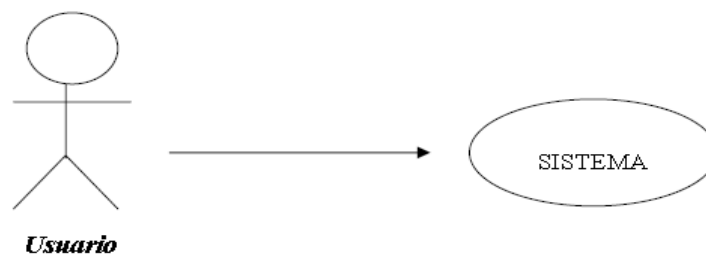


Ilustración 2: Caso de Uso (Sistema de Registro y Control) (Fuente Briones 2005).

Nombre Caso de Uso: General.

Actores: Suscriptor, Supervisor de Ventas, Supervisor de Calidad.

Descripción: Todos los usuarios tienen privilegios para entrar al sistema pero no tienen los mismos perfiles.

Resumen: Todos los usuarios tienen acceso al sistema, los usuarios administrativos tienen privilegio para todas las opciones, en cambio los suscriptores solo tienen para registrar y consultar (Ver Ilustración 3).

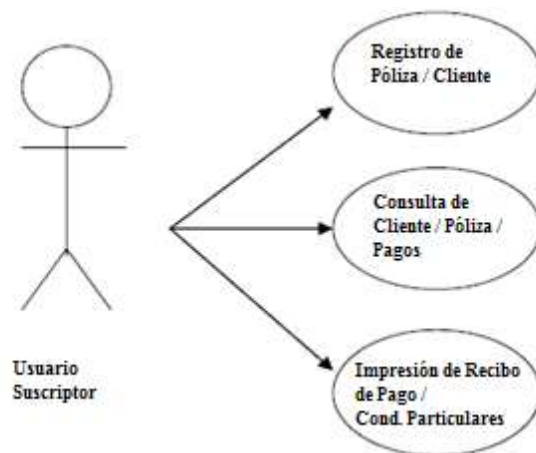


Ilustración 3: Caso de Uso General (Procesos a Realizar) (Fuente Briones 2005).

Nombre del Caso de Uso: Manejo de Clientes.

Actores: Suscriptor, Supervisor de Ventas.

Descripción: El suscriptor puede registrar nuevos clientes identificando toda la información de este el sistema debe validar que el código de cliente se encuentre disponible (no duplicado, que coincida con formato). También es posible modificar

algunas características del cliente y se puede eliminar clientes que no tengan otros registros que dependen de él (Ver Ilustración 4).

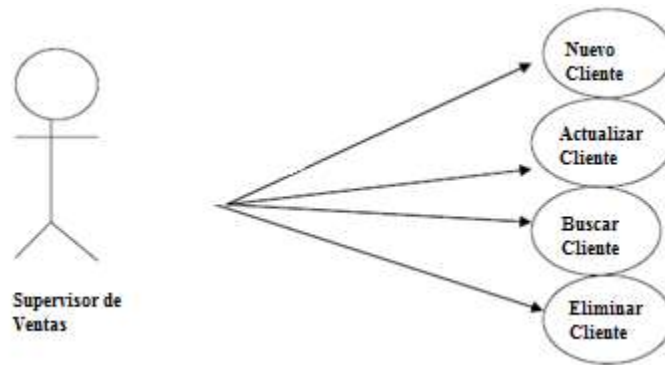


Ilustración 4: Caso de Uso (Manejo de Clientes) (Fuente Briones 2005).

Eventos:

Describimos los eventos que se llevarán a cabo para cada caso de uso relativo al manejo de clientes.

Nombre: Ingresar Clientes.

Actores: Suscriptor, Supervisor de Ventas.

Tabla 7: Ingresar Cliente.

Evento	Respuesta del Sistema
Selecciona Opción Ingreso de Clientes	Capturar Selección
	Mostrar Pantalla de Captura de Datos del Cliente
Digitar datos del nuevo cliente	Capturar datos del nuevo cliente
Selección de la opción guardar.	Valida datos del nuevo cliente

	Guarda datos del nuevo cliente
Eventos alternos	
El Cliente ya existe (ya existe el identificador)	No permite guardar y envía un mensaje de información
Deja algún dato en blanco	No permite guardar y envía un mensaje de información
No desea ingresar datos	Cierra la ventana de captura

Nombre: Modificar Clientes.

Actores: Suscriptor, Supervisor de Ventas.

Tabla 8: Modificar Cliente.

Evento	Respuesta del Sistema
Selecciona Opción Modificar Clientes	Capturar Selección
	Mostrar ventana con listado de Clientes existentes
Selecciona cliente a modificar	Capturar selección.
	Mostrar ventana de modificación de datos de cliente.
Modificar los datos deseados del cliente	Capturar datos modificados del cliente
Selección de la opción guardar	Valida datos del cliente modificado

	Guarda modificaciones a los datos del cliente
Eventos alternos	
El Cliente no existe (se modificó el identificador)	No permite modificar y envía un mensaje de información
Deja algún dato en blanco	No permite modificar y envía un mensaje de información
No desea modificar datos	Cierra la ventana de captura.

Nombre: Eliminar Clientes.

Actores: Supervisor de Ventas.

Tabla 9 : Eliminar Cliente.

Evento			Respuesta del Sistema
Selecciona	Opción	Eliminar	Capturar Selección
Clientes			Mostrar ventana con listado de Clientes existentes
Selecciona cliente a eliminar			Capturar selección. Mostrar de confirmación de eliminación de cliente.
Confirma eliminación de cliente			Capturar confirmación Eliminar cliente
Eventos alternos			

El Cliente no existe	No permite eliminar y envía un mensaje de información
Si el cliente tiene registros asociados	No permite eliminar y envía un mensaje de información
No desea eliminar cliente	Cierra la ventana de captura.

8.2 Diagrama de Actividades.

En esta etapa se muestran los diagramas de actividades que corresponden al caso de uso Manejo Cliente.

- Ingresar Cliente

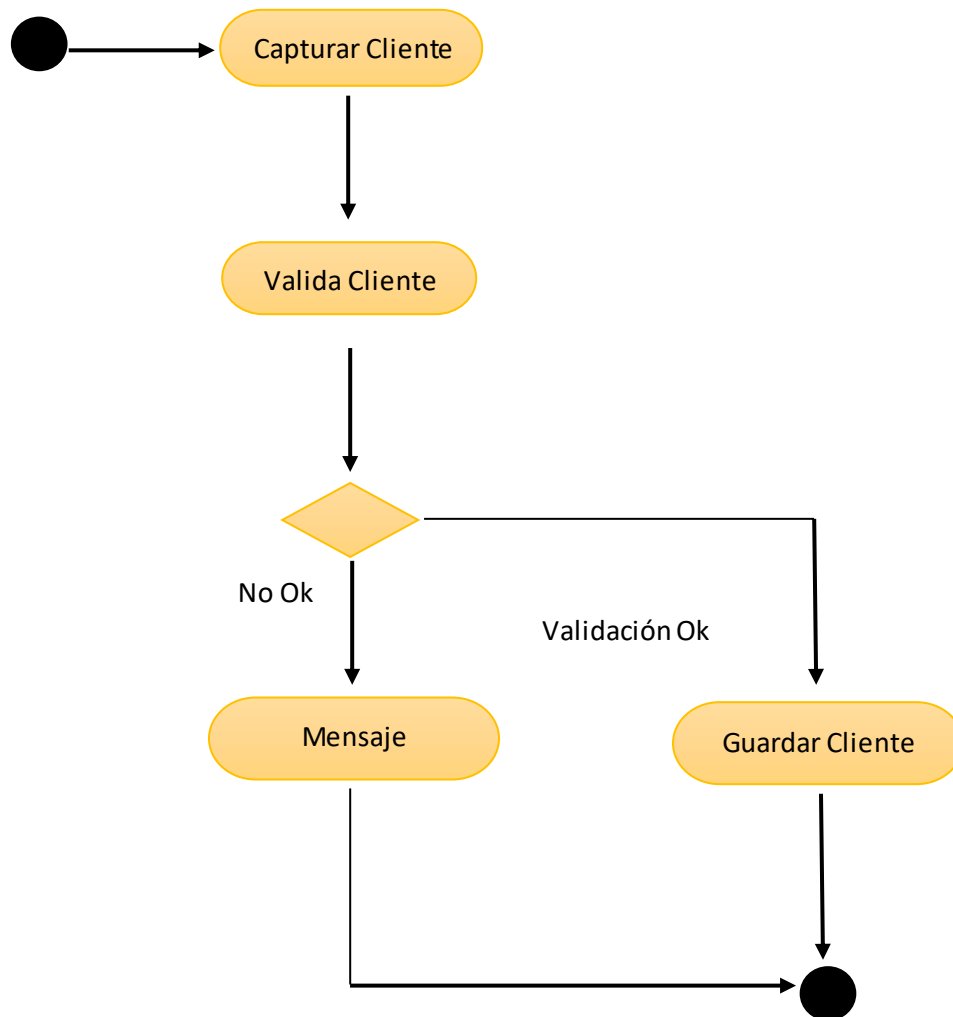


Ilustración 5: Ingreso de Cliente.

• Modificar Cliente

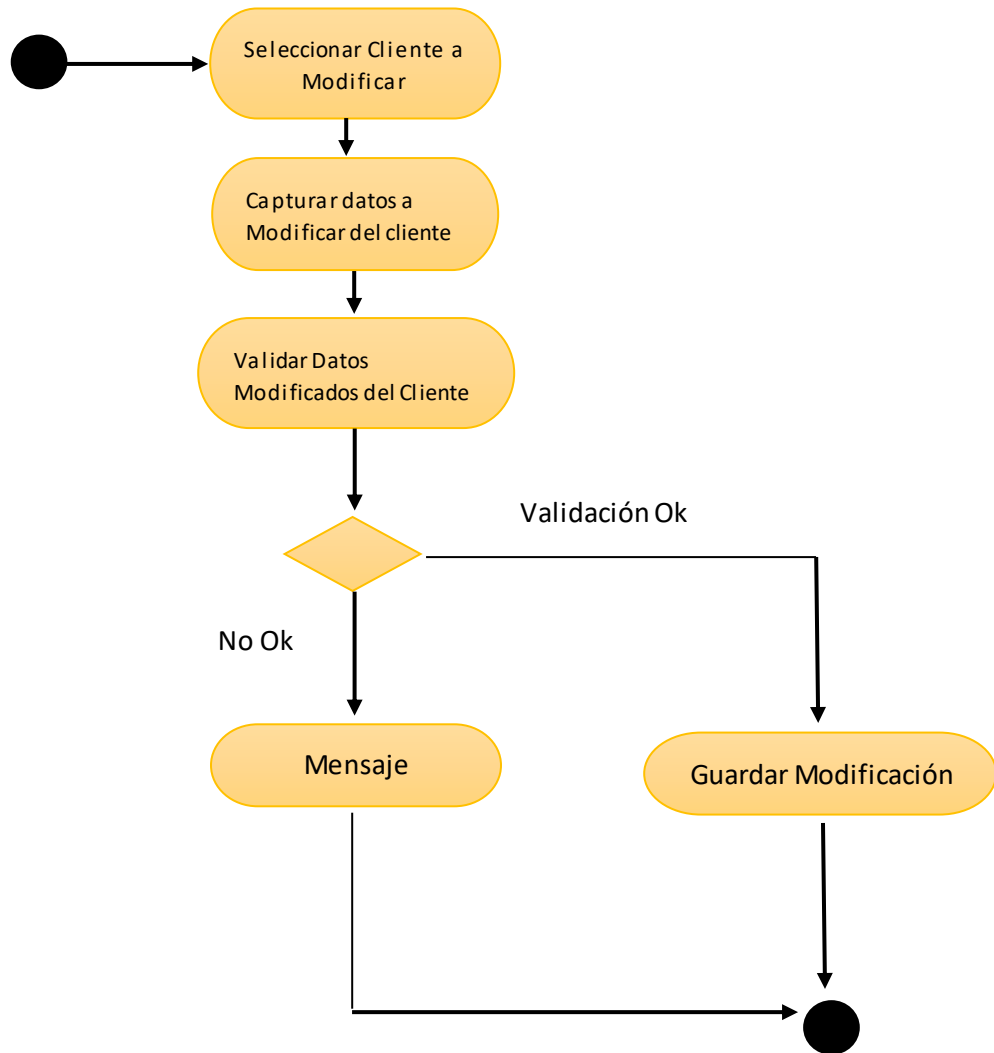


Ilustración 6: Modificar Cliente.

• Eliminar Cliente

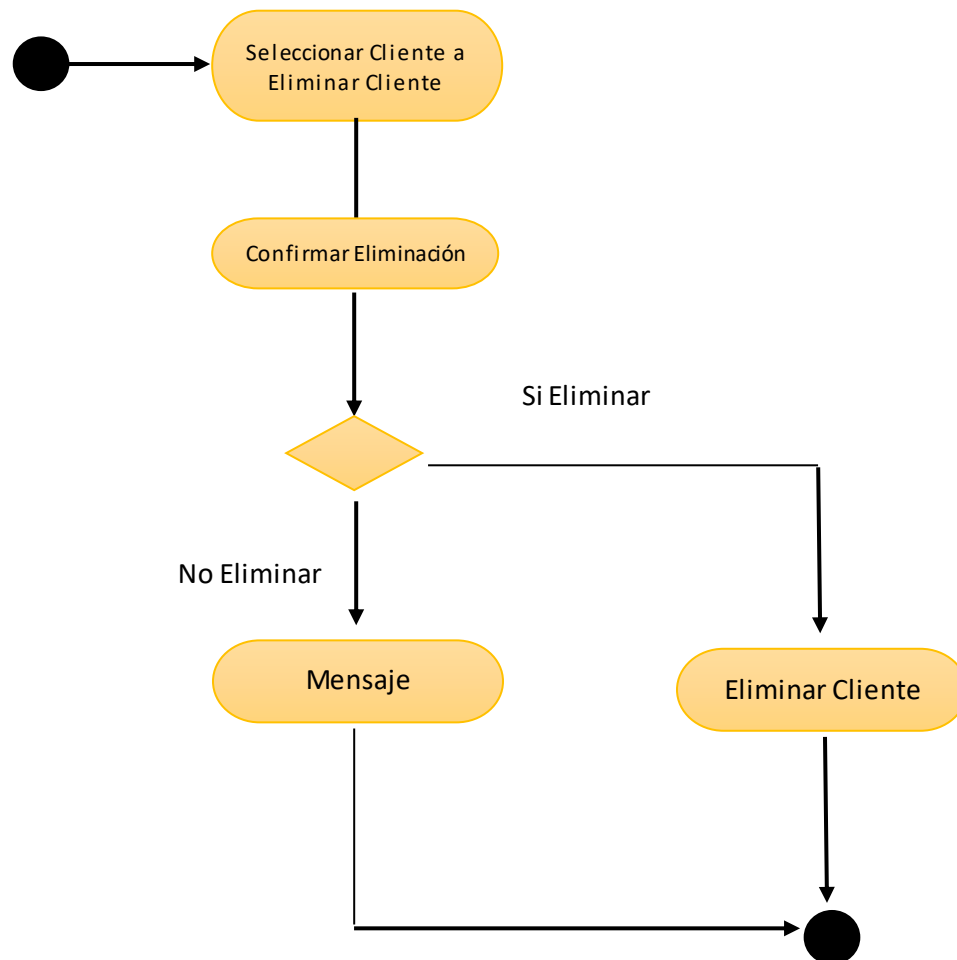


Ilustración 7: Eliminar de Cliente.

8.3 Desarrollo de Diagrama Entidad – Relación.

De la lectura y análisis detenido del ámbito del proyecto, fue posible extraer las siguientes entidades potenciales.

Tabla 10: Entidades Potenciales.

Aseguradoras	Administrador	Usuario
Corredurías	Inventario	Contador
Auditor	Pólizas	Software
Factura	Formatos	Equipos
Área de Ventas	Otros Ingresos	Devolución
Área Administradora	Vendedores	Sistema
Institución	Caja	Reclamo
SIBOIF		

Para analizar las entidades se debe identificar el tipo de información que pueden almacenar y determinar si esta información es necesaria o juega algún papel dentro del proceso que se automatizará. Primero se debe tener la idea del problema, preguntarse: ¿Qué se controlará?, ¿Quién lo hará?, ¿Cómo lo hace?, ¿Qué necesita?, etc. Se debe ubicar el analista dentro de la empresa e iniciar a dar forma, de manera global, a la solución del problema y los componentes que necesita. Con el tiempo y la práctica, este proceso se vuelve algo muy intuitivo.

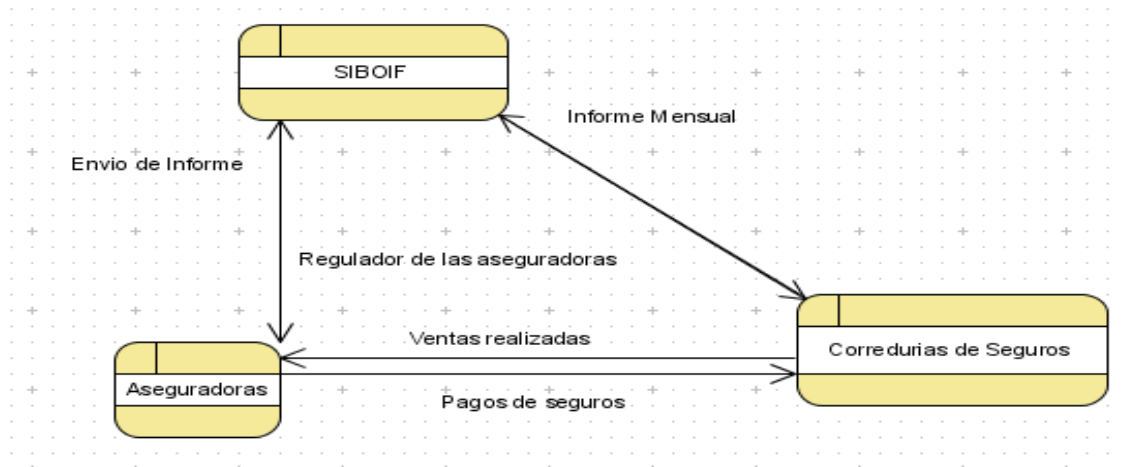
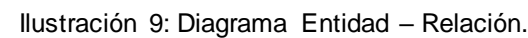


Ilustración 8: Diagrama Inicial con Todas las Entidades Resultantes de los Requerimientos.

Se muestra a continuación el diagrama final, el cual fue obtenido mediante normalización directa, llegando hasta la tercera forma normal.



9. DISEÑO DE LOS FORMULARIOS DEL SISTEMA.

Una vez desarrollado un análisis previo, entre dominios y requerimientos, se definen las características del sistema para generar una solución de software. Tomando como parámetros los requerimientos y el proceso de registro y control, con ayuda de una herramienta de programación se diseña el sistema, partiendo de la interfaz del usuario.

La primera interfaz a visualizarse (Ver Ilustración 10) es la que realiza una petición de Usuario y clave por dos líneas identificadas, en donde al ingresar los datos, de forma correcta y presionando el botón Entrar se apertura el acceso a las funciones del sistema.

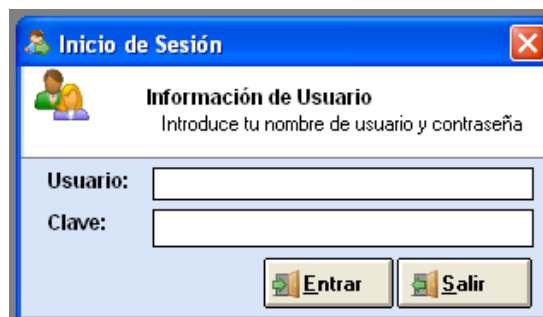


Ilustración 10: Ingreso de Clave de Acceso.

Luego de ingresar la clave se solicitara gravar la tasa de cambio del día (Ver Ilustración 11), que será utilizado para las conversiones de moneda que por el momento se manejan dólares y córdobas, seguido se mostrara la pantalla principal del sistema compuesta por una barra horizontal conteniendo las opciones de menú Datos Generales, Transacciones, Listados, Reportes, Herramientas, Ver, Ventana y Ayuda. Así como también una barra vertical de navegación rápida compuesta de los menús Movimientos, Catálogos Generales, Herramientas compuestas a su vez de submenús.

Una vez ingresado al sistema se solicitara gravar la tasa de cambio del día, que será utilizado para las conversiones de moneda que por el momento se manejan dólares y córdobas.



Ilustración 11: Ingreso de Tasa de Cambio del día.

La opción transacciones (Ver Ilustración 12), despliega un submenú que contiene las opciones de Póliza Automotor, Póliza Licencia, Abonos de Pólizas de Crédito y Modulo de Tipo de Documento. Cada una le permite al usuario entrar a las interfaces que le va a permitir Agregar, Actualizar, Eliminar y Buscar los respectivos datos según la opción a escoger.

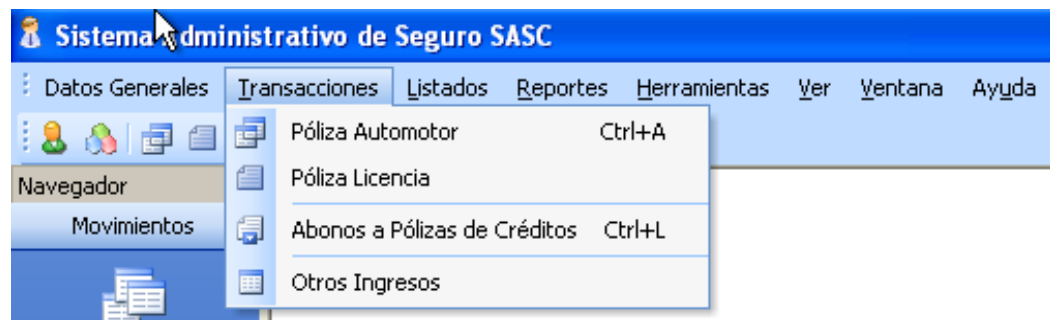


Ilustración 12: Submenú Transacciones

La opción de Póliza Automotor (Ver Ilustración 13), le permite al usuario introducir todos los datos concernientes a una póliza de automotor para posteriormente guardarlo en la base de datos.

Ilustración 13: Pantalla de Ingreso de Póliza Automotor.

La Opción de Póliza Licencia (Ver Ilustración 14), le permite al usuario introducir todos los datos concernientes a una póliza de licencia de conducir para posteriormente guardarlo en la base de datos.

Ilustración 14: Pantalla de Ingreso de Póliza Responsabilidad Civil de Licencia de Conducir.

La opción de Abonos a Pólizas de Créditos (Ver Ilustración 15), permite al usuario gravar los pagos a determinada póliza ya sea de Contado o de Crédito este último permite realizar múltiples pagos a una misma póliza dependiendo de la cantidad de cuotas.

Recibo de Pagos o Adelantos

Abono a Crédito
Permite modificar el monto de la deuda del cliente

Código Cliente

Número de Recibo: Fecha: 14/11/2016 Número de Póliza: 🔍

Tipo de Póliza: Hora:
Nombre: Monto Total:
Vendedor: Saldo:
Vence: Nuevo Vencimiento: ☐ Pagado con Capital Social

Pagos Pendientes para la Poliza Seleccionada						
	Cuota	CodCliente	NumFactura	Fecha	Monto	Interes

Concepto:

Monto a Pagar:

Ilustración 15: Pantalla de Ingreso de Abonos a Pólizas.

10. PRUEBAS DEL SISTEMA.

La fase de pruebas de la aplicación corresponde a las pruebas funcionales, pruebas de comportamiento y, por supuesto, las pruebas de aceptación por parte de los usuarios.

En esta etapa se midieron los logros en cuanto a la calidad de la aplicación realizada y la satisfacción que esta ofrece a los cliente, para ello se analizaron sus factores de éxito a partir de los objetivos y la justificación, encontrándose que para ser exitosa debe de pasar con éxito las siguientes pruebas:

Pruebas Funcionales: Son las pruebas relacionadas con la funcionalidad de la aplicación.

Pruebas de Comportamiento: Esa prueba evalúa como la aplicación obtiene los resultados, más no cuales son los resultados obtenidos, además de evaluar el manejo de concurrencias, el estrés, el rendimiento y la velocidad.

Pruebas de Aceptación: Esta prueba se refiere a la aceptación de la aplicación por parte de los usuarios y así determinar si los requerimientos de los mismos son los obtenidos en el análisis de requerimientos.

Tabla 11: Pruebas Funcionales				
Tipo de Pruebas	Actividades	Autores	Técnica	Resultados
Pruebas del Sistema	Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación.	<ul style="list-style-type: none"> • Usuarios del sistema • Desarrolladores • DBA 	<ul style="list-style-type: none"> • ¿Los resultados esperados ocurren cuando se utiliza un dato válido? • ¿Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido? • 	<ul style="list-style-type: none"> • Si, los usuarios demostraron que el ingresar datos validos al sistema este respondió de la manera esperada. • Si, cuando el usuario introduce un dato errado o incorrecto las rutinas de excepciones le especifica al usuario el error correspondiente
Prueba de desempeño	Comprobar el tiempo de respuesta para las transacciones o funciones de negocios	<ul style="list-style-type: none"> • Cliente • Usuarios del Sistema • Desarrolladores 	<ul style="list-style-type: none"> • Comparar el desempeño del sistema actual con los procedimientos anteriores. 	<ul style="list-style-type: none"> • La utilización del sistema aumento el desempeño al agilizar la velocidad de búsqueda y registro de los clientes y sus respectivas pólizas, ya que los procedimientos utilizados anteriormente eran manuales.

Pruebas de Carga	Verificar el tiempo de respuesta del sistema para transacciones o casos de uso de negocios, bajo diferentes condiciones de carga.	<ul style="list-style-type: none"> • Usuarios del sistema • Desarrolladores • DBA 	<ul style="list-style-type: none"> • Usar los scripts desarrollados para Pruebas del Negocio. • Modificar archivos de datos (para incrementar el número de transacciones o veces que cada transacción ocurre). 	<ul style="list-style-type: none"> • Se utilizó un procedimiento almacenado que permitió crear cargas de múltiples tracciones al mismo tiempo. El sistema respondió correctamente y no hubo ningún error no controlado.
Pruebas de Integridad de Datos y Base de Datos	Asegurar que los métodos de acceso y procesos funcionan adecuadamente y sin ocasionar corrupción de datos.	<ul style="list-style-type: none"> • Desarrolladores • DBA 	<ul style="list-style-type: none"> • Invocar cada método de acceso y proceso de la Base de datos, utilizando en cada uno datos válidos e inválidos. • Analizar la Base de datos, para asegurar que los datos han sido grabados apropiadamente, que todos los eventos de Base de datos se ejecutaron en forma correcta y revisar los datos retornados en 	<ul style="list-style-type: none"> • Se utilizaron los procedimientos almacenados y disparadores correspondientes en SQL Server 2005 para comprobar que los datos introducidos fueron grabados apropiadamente.

			diferentes consultas.	
Pruebas de Seguridad y Control de Acceso	Verificar que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido.	<ul style="list-style-type: none"> • Usuarios del Sistema • Desarrolladores 	<ul style="list-style-type: none"> • Comprobar la seguridad del sistema, incluyendo acceso a datos o Funciones de negocios • Comprobar la seguridad de la Base de Datos, incluyendo ingresos y accesos remotos al sistema. 	<ul style="list-style-type: none"> • Se comprobó que por medio de los permisos otorgados a cada usuario, según su función pudieron manipular el sistema. • El acceso a la base de datos solo es otorgado al DBA, responsable del mantenimiento y respaldo de la misma.
Tabla 12: Pruebas de Comportamiento				
Pruebas de Stress	Verificar que el sistema funciona apropiadamente y sin errores, bajo condiciones de stress	<ul style="list-style-type: none"> • Desarrolladores • DBA 	<ul style="list-style-type: none"> • ¿Memoria baja o no disponible en el servidor? 	<ul style="list-style-type: none"> • Siendo este un sistema para una pequeña cantidad de usuarios y transacciones, la memoria disponible en el servidor es suficiente para la cantidad de datos que ocupa la aplicación y el Gestor de Bases de Datos Ms SQL Server 2005

			<ul style="list-style-type: none"> • ¿Máximo número de clientes conectados o simulados (actuales o físicamente posibles)? • ¿Múltiples usuarios desempeñando la misma transacción con los mismos datos? 	<ul style="list-style-type: none"> • La cantidad máxima de personas conectadas de forma simultánea son 10. • El sistema soporta múltiples usuarios realizando las mismas transacciones al mismo tiempo.
Pruebas de Volumen	Las pruebas de volumen hacen referencia a grandes cantidades de datos para determinar los límites en los cuales el Sistema falle.	<ul style="list-style-type: none"> • Desarrolladores • DBA 	<ul style="list-style-type: none"> • Determinar el Máximo (actual o físicamente posible) número de clientes conectados (o simulados), todos ejecutando la misma función (peor caso de desempeño) por un período extendido. • Determinar el Máximo tamaño de la base de datos (actual o escalado) y múltiples consultas 	<ul style="list-style-type: none"> • El máximo de usuarios conectados realizando la misma transacción, por un periodo mayor a 8 horas son 5. • El tamaño máximo de la base de datos es 524.272 terabytes y de 32 bytes por consultas simultaneas.

			ejecutadas simultáneamente	
Tabla 13: Pruebas de Aceptación				
Pruebas del Ciclo del Negocio	Asegurar que el sistema funciona de acuerdo con el modelo de negocios emulando todos los eventos en el tiempo y en función del tiempo.	<ul style="list-style-type: none"> • Cliente • Desarrolladores 	<ul style="list-style-type: none"> • Todas las fechas o funciones que involucren tiempos serán probadas con datos válidos e inválidos de fechas o periodos de tiempo. • Todas las funciones ocurren en un periodo de tiempo serán ejecutadas en el tiempo apropiado. • Cada regla de negocios es aplicada adecuadamente. 	<ul style="list-style-type: none"> • Las funciones que involucran tiempo son aquellas que están relacionadas con las salidas y reportes del sistema. Como este es generado de manera automatizado los tiempos de espera son mínimos. • Todas las funciones y procedimientos del sistema se hacen en el tiempo que son requeridos. • Todas las normas y reglamentos ya sean propios de la empresa, así como aquellas que son de la SIBOIF son aplicadas en los procedimientos y salidas del sistema

Pruebas de GUI	<p>Comprobar que las navegaciones a través de los objetos de la prueba reflejan las funcionalidades del negocio y requisitos, se realiza una navegación ventana por ventana, usando los modos de acceso (tabuladores, movimientos del mouse, teclas rápidas, etc)</p>	<ul style="list-style-type: none"> • Usuarios de Sistema • Desarrolladores 	<ul style="list-style-type: none"> • Pruebas de crear /modificar cada ventana para verificar la adecuada navegación y estado de los objetos. 	<ul style="list-style-type: none"> • Se comprobó la navegación de las pantallas utilizando los diferentes métodos de entrada, siendo satisfactorias al desplazarse correctamente por cada campo.
Pruebas de Configuración	<p>Validar y verificar que el “Usuario del Sistema” funciona apropiadamente en las estaciones de trabajo recomendadas.</p>	<ul style="list-style-type: none"> • Usuarios de Sistema • Desarrolladores 	<ul style="list-style-type: none"> • Incluir la apertura o cierre de varias aplicaciones Microsoft, como Excel y Word (o algún tipo de software similar a la que se está probando) como una parte de la prueba, ya sea al comienzo o en 	<ul style="list-style-type: none"> • Tanto el sistema como los diferentes programas instalados e utilizados en los equipos funcionan e interactúan de forma correcta, por lo que no se afectan entre sí, ni a la funcionalidad que el cliente

			algún momento intermedio.	desempeña de forma cotidiana.
Prueba de Aceptación	Determinar por parte del cliente la aceptación o rechazo del sistema desarrollado.	<ul style="list-style-type: none"> • Cliente • Usuarios de Sistema • Desarrolladores • DBA 	<ul style="list-style-type: none"> • La prueba de aceptación es ejecutada antes de que la aplicación sea instalada dentro de un ambiente de producción. 	<ul style="list-style-type: none"> • Antes de la instalación del sistema se procedió a realizar una serie de pruebas (Ver las pruebas anteriores) donde se validó con el cliente y/o usuarios toda la funcionalidad del sistema, satisfaciendo todos los requisitos previos del cliente.
Prueba de Instalación	Verificar y validar que el sistema se instala apropiadamente en cada cliente, bajo las siguientes condiciones:	<ul style="list-style-type: none"> • Usuarios de Sistema • Desarrolladores • DBA 	<ul style="list-style-type: none"> • Instalaciones nuevas, nuevas máquinas a las que nunca se les ha instalado el sistema. • Actualizar máquinas previamente instaladas con el sistema. 	<ul style="list-style-type: none"> • Se instaló de forma correcta y rápida el ejecutable del sistema en los nuevos equipos. • Se instaló de forma correcta y rápida el ejecutable del sistema en máquinas ya en uso.

11. CONCLUSIÓN

Se logró diagnosticar la situación actual desde el punto de vista cualitativo y cuantitativo de las corredurías. Utilizando entrevistas y encuestas y a través de estas lograr determinar los requerimientos que debía cumplir la aplicación.

El sistema es un ejemplo de lo útil y eficiente que son las herramientas y estándares abiertos para el desarrollo de aplicaciones de escritorio, así como el beneficio económico que estas proporcionan, demuestra que pueden construirse aplicaciones de bajo costo que tienen las ventajas y beneficios que presentan las herramientas comerciales.

El factor tiempo es en el que se logra más beneficio, ya que las corredurías pueden cumplir con sus informes al momento de ser solicitado, sin el mínimo retraso.

Los Sistemas de Información, han alcanzado sin duda alguna en los últimos años un gran auge en el desarrollo de las distintas áreas de trabajo debido a que su utilización agiliza los procesos, proporcionando información altamente confiable y en muchos casos garantizando la seguridad tanto de los datos a procesar como a la información.

12. RECOMENDACIONES

Se recomienda, capacitación para todo el personal involucrado; de este modo se asegura el uso de la herramienta para la productividad organización y al mismo tiempo se presenta como insumos para evidencia el correcto funcionamiento o posibles ajustes que se deban realizar.

Otra recomendación sería incluir más módulos, para que el sistema se adapte a las necesidades según la oferta de nuevos productos de las aseguradoras y la demanda de los clientes, contribuyendo a un aumento en las ventas e ingreso de la correduría.

13. BIBLIOGRAFÍA

- Aranda, G. (Junio de 2016). *Microsoft SQL Server*. Obtenido de <https://prezi.com/0m9hgiostlsf/microsoft-sql-server/>
- Ary, S. (16 de Marzo de 2003). <http://docente.ucol.mx>. Obtenido de <http://docente.ucol.mx>: http://docente.ucol.mx/sadanary/public_html/bd/cs.htm
- Aydoo, C. (Mayo de 2015). <http://profesores.fi-b.unam.mx>. Obtenido de <http://profesores.fi-b.unam.mx>: <http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html>
- Busot, A. (1991). *Investigación Educacional*. Maracaibo: Luz.
- Carvajalino, J. (25 de Mayo de 2015). <http://carvajalino09.blogspot.com>. Obtenido de <http://carvajalino09.blogspot.com>: <http://carvajalino09.blogspot.com/2015/05/caracteristicas-de-visual-basic.html>
- Cohen, D. y. (2000). *Sistemas de Información para los negocios*. Mexico: McGraw-Hill.
- Diaz, J. (19 de Abril de 2015). Obtenido de <https://msdn.microsoft.com>: <https://msdn.microsoft.com/es-es/library/ms165088%28v=vs.100%29.aspx>
- Humboldt, U. A. (28 de 03 de 2015). *Ingeniería del Software UAH 2015-I*. Obtenido de <https://ingenieriadelsoftwareuah2015.wordpress.com/2015/03/28/meto-do-blue-watch/>
- jiparra. (28 de Agosto de 2008). <https://jiparra.wordpress.com>. Obtenido de <https://jiparra.wordpress.com/2008/08/28/caracteristicas-net/>
- Lopez Carrasco, A. (3 de octubre de 2011). <https://docs.google.com>. Obtenido de <https://docs.google.com>: https://docs.google.com/document/d/1Kpei0XxuJBIII0T4XbpcfWY0LHcG-0g9kKXb9TGqvRc/edit?hl=en_US&pref=2&pli=1
- Molina, L. (Marzo de 2013). <https://laurmolina7821.wordpress.com>. Obtenido de <https://laurmolina7821.wordpress.com>: <https://laurmolina7821.wordpress.com/1-1-2-aplicaciones-cliente-servidor/>
- Montilva, J. (2000). *Modelo de procesos para el desarrollo de software*. Venezuela: Consejo de publicaciones de la Universidad de los Andes.
- Provencio, F. R. (10 de Diciembre de 2003). Obtenido de <http://www.desarrolloweb.com>: <http://www.desarrolloweb.com/articulos/1329.php>

- Tamayo, M. T. (30 de Agosto de 2002). *El proceso de la Investigación*. Mexico: Limusa S.A. Obtenido de http://www.academia.edu/6609429/UNIDAD_I_TIPOS_DE_INVESTIGACION
- Valenzuela, M. (09 de 10 de 2013). <http://www.laprensa.com.ni>. Obtenido de <http://www.laprensa.com.ni/2013/10/09/economia/165369-detras-de-los-seguros>
- Viera, K. R. (30 de Mayo de 2014). <http://es.slideshare.net>. Obtenido de <http://es.slideshare.net: http://es.slideshare.net/guest1675627/materia-clienteservidor>

14. ANEXOS

Plan de Trabajo

Nombre de tarea ▼	Duración ▼	Predecesoras ▼	Nombres de los recursos ▼
▣ Desarrollo del Sistema SASC	130 días		Jose Fonseca, Christian Rocha
▣ INICIO DEL PROYECTO	130 días		
▷ DEFINICIÓN DE REQUISITOS Y REQUERIMIENTOS	4 días		
Fin de definición de requisitos y Requerimientos			
▣ ANÁLISIS	22 días		
Determinacion del alcance del sistema			
Identificacion del entorno del sistema			
Identificacion del entorno Tecnológico			
Analisis de requisitos del sistema			
Reunion de Seguimiento			
Identificacion de Subsistema de analisis			
Elaboracion del Modelo de Datos conceptual			
Elaboracion del Modelo de Datos Logico			
Analisis de Consistencia y especificacion de requisitos			
Reunion de Seguimiento			
Fin Análisis			
▷ DISEÑO	24 días		
Fin de Diseño			
▷ IMPLEMENTACIÓN Y EVALUACIÓN	110 días		
Fin de Implementación y Evaluación			
▷ IMPLANTACIÓN	15 días		
Fin de Implantación			
FIN DEL PROYECTO.			

15. FORMATO DE ENCUESTA

Nombre del Puesto: _____

Representante: _____

Fecha:

Preguntas

1.- ¿En cuánto tiempo registra la información de un cliente?

- a. 5 min.
- b. 10 min.
- c. 15 min.
- d. Otros. Especifique_____.

2.- ¿En cuánto tiempo registras la información de una póliza?

- a. 5 min.
- b. 10 min.
- c. 15 min.
- d. Otros. Especifique_____.

3.- ¿En cuánto tiempo registras la información de un reclamo?

- a. 5 min.
- b. 10 min.
- c. 15 min.
- d. Otros. Especifique_____.

4.- ¿Cuánto tiempo le toma elaborar y entregar el informe de ventas?

- a. 5 min.
- b. 10 min.

c. 15 min.

d. Otros. Especifique_____.

5.- ¿Cuántas veces tiene que repetir la elaboración y entrega de los informes de ventas?

a. 1 vez.

b. 2 veces.

c. 3 veces.

d. Otros. Especifique_____.

6.- ¿Cuánto tiempo le toma elaborar y entregar los informes internos?

a. 5 min.

b. 10 min.

c. 15 min.

d. Otros. Especifique_____.

7.- ¿Cuántas veces tiene que repetir la elaboración y entrega de los informes internos?

a. 1 vez.

b. 2 veces.

c. 3 veces.

d. Otros. Especifique_____.

7.- ¿Cuánto tiempo le toma elaborar y entregar los informes internos?

a. 5 min.

b. 10 min.

c. 15 min.

d. Otros. Especifique_____.

8.- Indique con una (X) cuales beneficios del Sistema de Información

Automatizado considera relevante para usted.

a. Mayor Seguridad de la Información

b. Rapidez en el acceso datos

c. Mayor control sobre los mismos

d. Otros

16. EJEMPLOS DE PÓLIZAS.

INISER
Seguridad que le da confianza

INSTITUTO NICARAGÜENSE DE SEGUROS Y REASEGURO
Km. 4 1/2 Carretera Sur Managua, Nicaragua
PBX: (505) 2255-7575, Fax: 2255-7571
Web site: www.iniser.com.ni E-mail: iniser@iniser.com.ni

PÓLIZA DE RESPONSABILIDAD CIVIL DE LICENCIA ORDINARIA

CONDICIONES PARTICULARES

INI - 390

PÓLIZA No. **1012915** CERT. N° **1** Vigencia desde: **08/11/12** hasta: **07/11/13**

Nombre del Asegurado: **JUAN CARLOS GUTIERREZ GAITAN** Cédula: **001-0302780031-W**

Dirección: **ULTANIC** Teléfono: **2284144**

Municipio: **MANAGUA** Departamento: **MANAGUA** Nacionalidad: **NICARAGUA**

COBERTURAS	SUMAS ASEGURADAS
a) Muerte o lesiones a una persona	C\$ 5000
b) Muerte o lesiones a dos o más personas	C\$ 10000
c) Daños a bienes de terceros	C\$ 5000

DATOS DE LA LICENCIA DEL ASEGURADO

LICENCIA No. **001-0302780031-W** FECHA DE EXPIRACIÓN DE LA LICENCIA **03-02-2014**

CATEGORÍAS ☒ ☒ ☒

Observaciones: **08-NOV-1210:06:14 AM**

Firma y sello del Cajero

Firma y sello autorizado

Nota: Esta póliza no tendrá valor si no refleja la firma y el sello del funcionario autorizado del INISER.

Póliza emitida por: Intermediario No. **000000**

Forma de pago: Efectivo ☐ Ck No. Banco

Tarjeta de crédito No.: PRIMA TOTAL C\$ **170.00**

Original: Cliente **Clausula y Condiciones Generales (Ver reverso)**

AVISO DE SINISTRO

Dentro de las tres días siguientes al siniestro, el asegurado debe avisar al INISER y presentar certificado a falta de las autoridades correspondientes, en el que constare su intervención. La falta de aviso dentro del período señalado librará al INISER de toda responsabilidad.

Al momento de un accidente llamar al teléfono: **2255-7575/Ext. 3803-3818**

Firma Autorizada

INISER
INSTITUTO NICARAGÜENSE DE SEGUROS Y REASEGURO
PBX: (505) 2255-7575, Fax: (505) 2255-7571
Web site: www.iniser.com.ni E-mail: iniser@iniser.com.ni

PÓLIZA No. **1012915** VENCE: **07/11/13**

SEGURO DE RESPONSABILIDAD CIVIL DE LICENCIA ORDINARIA

ASEGURADO: **JUAN CARLOS GUTIERREZ GAITAN**

CÉDULA No.: **001-0302780031-W**

DIRECCIÓN: **ULTANIC**

LICENCIA No.: **001-0302780031-W**

CATEGORÍA: ☒ ☒ ☒



INSTITUTO NICARAGÜENSE DE SEGUROS Y REASEGUROS
Km. 4½ Carretera Sur, Managua Nicaragua
PBX: (505) 2255-7575, Fax: (505) 2255-7570
Web site: www.iniser.com.ni E-mail: iniser@iniser.com.ni

PÓLIZA No.: **010000**
3

**SEGURO OBLIGATORIO DE RESPONSABILIDAD CIVIL
POR DAÑOS A TERCEROS PARA VEHÍCULOS CON MATRÍCULA EXTRANJERA**

CONDICIONES PARTICULARES

Plan VEHICULO EXTRAJERO	Vigencia: Desde 12:03:00 15/03/2013	Hasta 24:00 Hrs 14/04/2013
Nombre del Asegurado	MARIO ERNESTO REYES ZUNIGA	
Cédula o Pasaporte No.	0035011977060930	Teléfono: 50522798634, 50587180611
Nacionalidad	HONDURAS	
Motivo de Ingreso	Interés asegurable: Propietario <input type="checkbox"/> Conductor <input checked="" type="checkbox"/>	
Puesto Fronterizo:	Tránsito <input type="checkbox"/> Residencia <input type="checkbox"/> Estudios <input type="checkbox"/> Negocio <input checked="" type="checkbox"/> Turismo <input type="checkbox"/> Peñas Blancas <input type="checkbox"/> El Guasaule <input checked="" type="checkbox"/> Las Manos <input type="checkbox"/> El espino <input type="checkbox"/> Otros <input type="checkbox"/>	
Sumas aseguradas	a) Muerte o lesiones a una persona US\$ 5,000.00 b) Muerte o lesiones a dos o más personas US\$ 10,000.00 c) Daños a bienes de terceros US\$ 5,000.00	

DATOS PARTICULARES DEL VEHÍCULO ASEGURADO

Placa	PBH0040	Pasajeros	4
Marca	NISSAN	Año	2000
Tipo	PICKUP	Color	AZUL CLARO
No. Chasis	3N6CD13S1ZK016473	Circulación No	198971

Observaciones 15-MAR-1309:42:42 A

Firma y sello del Cajero

Nota: Esta póliza no tendrá valor si no refleja la firma y el sello del funcionario autorizado de INISER.

Firma y sello autorizado



999999

Póliza emitida por: Intermediario No.

Forma de pago:

CONTADO

Efectivo Ck No. Banco

Tarjeta de crédito

Prima Total **C\$** **U\$** 12.00

INSTITUTO NICARAGÜENSE DE SEGUROS Y REASEGUROS	
Km. 4½ Carretera Sur, Managua Nicaragua	
PBX: (505) 2255-7575, Fax: (505) 2255-7570	
Web site: www.iniser.com.ni E-mail: iniser@iniser.com.ni	
PÓLIZA No. 010000	Certificado No. 1
VENGE: 14/04/2013	
SEGURO OBLIGATORIO DE RESPONSABILIDAD CIVIL POR DAÑOS A TERCEROS PARA VEHÍCULOS CON MATRÍCULA EXTRANJERA	
ASEGURADO	
MARIO ERNESTO REYES ZUNIGA	
CÉDULA o PASAPORTE No.	0035011977060930
SUMA ASEGURADA	Según Arto 69 Ley 431
CHASIS No.	3N6CD13S1ZK016473
MAIRCA	NISSAN
PLACA No.	PBH0040
Interés asegurable: Propietario <input type="checkbox"/> Conductor <input checked="" type="checkbox"/>	